
Entwicklung einer umfassenden Metrik für die Bewertung einer Lidar-Sensor- Simulation durch Betrachtung mehrerer aufeinander folgender Verarbeitungsebenen

Masterthesis Nr. 699/18

Bearbeiter: Sebastian Huch | 2254359

Betreuer: Philipp Rosenberger, M. Sc.



TECHNISCHE
UNIVERSITÄT
DARMSTADT



FAHRZEUGTECHNIK
TU DARMSTADT

Sebastian Huch
Matrikelnummer: 2254359
Studiengang: Master Maschinenbau

Masterthesis Nr. 699/18
Thema: Entwicklung einer umfassenden Metrik für die Bewertung einer Lidar-Sensor-Simulation
durch Betrachtung mehrerer aufeinander folgender Verarbeitungsebenen

Eingereicht: 08. November 2018

Technische Universität Darmstadt
Fachgebiet Fahrzeugtechnik
Prof. Dr. rer. nat. Hermann Winner
Otto-Berndt-Straße 2
64287 Darmstadt

Dieses Dokument wird bereitgestellt von TUPrints – Publikationsservice der TU Darmstadt.

<https://tuprints.ulb.tu-darmstadt.de/>

Bitte verweisen Sie auf:

URN: urn:nbn:de:tuda-tuprints-83970

URI: <https://tuprints.ulb.tu-darmstadt.de/id/eprint/8397>

Lizenz: CC-BY-NC-ND 4.0 International

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Erklärung

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 und § 23 Abs. 7 APB TU Darmstadt

Hiermit versichere ich, Sebastian Huch, die vorliegende Master-Thesis gemäß § 22 Abs. 7 APB der TU Darmstadt ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß § 23 Abs. 7 APB überein.

English translation for information purposes only:

Thesis Statement pursuant to § 22 paragraph 7 and § 23 paragraph 7 of APB TU Darmstadt

I herewith formally declare that I, Sebastian Huch, have written the submitted thesis independently pursuant to § 22 paragraph 7 of APB TU Darmstadt. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content. This thesis has not been handed in or published before in the same or similar form.

I am aware, that in case of an attempt at deception based on plagiarism (§38 Abs. 2 APB), the thesis would be graded with 5,0 and counted as one failed examination attempt. The thesis may only be repeated once.

In the submitted thesis the written copies and the electronic version for archiving are pursuant to § 23 paragraph 7 of APB identical in content.

Datum/Date: 08. November 2018

Unterschrift/Signature:

Zusammenfassung

Mit der Entwicklung von neuen hochautomatisierten Fahrfunktionen (HAF) steigen die Anforderungen an deren simulationsbasierten Absicherung. Hierzu werden im Rahmen der Forschungsprojekte PEGASUS und ENABLE-S3 am Fachgebiet Fahrzeugtechnik (FZD) der Technischen Universität Darmstadt Sensormodelle für aktive Sensoren wie Lidar-, Radar- und Ultraschallsensoren entwickelt und validiert. Diese Sensormodelle ermöglichen eine Steigerung der Effizienz bei der Freigabe hochautomatisierter Fahrfunktionen.

In der vorliegenden Masterthesis wird eine Methodik entwickelt, die mittels Metriken eine ganzheitliche Bewertung des Lidar-Sensormodells und den anschließenden Verarbeitungsebenen ermöglicht. Die Anwendung der Methodik wird anhand beispielhafter Szenarien gezeigt.

Zunächst werden mögliche Fehler in der Lidar-Sensorsimulation mit einer FMEA (Fehlzustandsart- und -auswirkungsanalyse) analysiert. Dabei ist das primäre Ziel, zu überprüfen, ob die Anwendung einer FMEA zur Bewertung einer Lidar-Sensorsimulation möglich ist. Die beispielhafte FMEA zeigt, dass für eine vollständige FMEA mehr Informationen über das Zusammenwirken der Verarbeitungsebenen benötigt werden. Die Zusammenhänge zwischen den Verarbeitungsebenen werden jedoch ersichtlich.

Die entwickelte Methodik zur Bewertung der einzelnen Verarbeitungsebenen sieht vor, zuerst reale Messdaten in definierten Szenarien aufzuzeichnen. Mit den Informationen aus den realen Versuchen werden die Szenarien im Simulationstool VTD nachgestellt. Der Ausgang der Simulation ist die erste Verarbeitungsebene „Punktwolken“, der anschließend in den Verarbeitungsebenen „Segmentierung“, „Tracking“ und „Klassifizierung“ verwendet wird. Für alle Verarbeitungsebenen werden Metriken recherchiert, die einen Vergleich der jeweiligen Verarbeitungsebenen zwischen den realen und simulierten Daten ermöglichen. Außerdem werden Überlegungen zur Erstellung einer Gesamtmetriken angestellt. Da diese jedoch nicht physikalisch interpretierbar ist und Informationen über die Fehler in den einzelnen Verarbeitungsebenen verloren gehen, wird keine Gesamtmetriken erstellt.

Mittels beispielhafter statischer und dynamischer Szenarien erfolgt die Bewertung der Verarbeitungsebenen „Punktwolken“ und „Tracking“ sowie die Überprüfung der Anwendbarkeit der Metriken. Für die Bewertung der Verarbeitungsebenen „Punktwolken“ werden Occupancy Grids erstellt. Die Sensibilität der Metriken wird überprüft, indem Parameter variiert und Szenarien manuell manipuliert werden. Da keine Erfahrungswerte vorliegen, bei welchen Werten der Metriken die Simulationsqualität als hoch eingestuft werden kann, können die Metriken nur für den Vergleich unterschiedlicher Szenarien und unterschiedlicher Simulationen eingesetzt werden.

In einer abschließenden kombinierten Betrachtung der Verarbeitungsebenen „Punktwolken“ und „Tracking“ werden bei einem Szenario für jeden Zeitschritt die Metriken beider Verarbeitungsebenen berechnet und die Korrelation zwischen den Verarbeitungsebenen überprüft. Die Verarbeitungsebenen zeigen eine hohe Korrelation, was darauf schließen lässt, dass die Qualität der Punktwolken der Lidar-Sensorsimulation einen großen Einfluss auf die Ergebnisse nachgelagerter Verarbeitungsebenen wie beispielsweise das Tracking hat.

Inhaltsverzeichnis

Zusammenfassung.....	I
Inhaltsverzeichnis.....	II
Formelzeichen und Indizes	IV
Abkürzungsverzeichnis.....	VII
Abbildungsverzeichnis.....	IX
Tabellenverzeichnis	XI
1 Einleitung	1
1.1 Motivation.....	1
1.2 Konkretisierung der Aufgabenstellung	2
1.3 Methodik des Vorgehens	3
2 Theoretische Grundlagen	4
2.1 Der Lidar-Sensor.....	4
2.1.1 Aufbau und Funktionsprinzip des Lidar-Sensors.....	4
2.1.2 Beschreibung der Lidar-Gleichung	6
2.1.3 Spezifikationen des ibeo Lux 2010®	8
2.2 Sensoren in der Simulation	10
2.3 Lidar-Sensorsimulation bei FZD	13
2.3.1 Segmentierung	14
2.3.2 Tracking	18
2.3.3 Klassifizierung	19
3 Analyse möglicher Fehler in der Lidar-Sensorsimulation mittels FMEA	22
4 Übersicht Validierungsmethoden und angewandte Methodik	28
5 Metriken zur Bewertung der Verarbeitungsebenen	33
5.1 Definition und Abgrenzung der Verarbeitungsebenen	33
5.2 Punktwolken.....	33
5.3 Segmentierung	40
5.4 Tracking	41
5.4.1 Hausdorff-Metrik	42
5.4.2 OMAT-Metrik.....	42
5.4.3 OSPA-Metrik	44
5.4.4 OSPA-T-Metrik	44
5.4.5 GOSPA-Metrik	46
5.4.6 Rahmathullah et al.-Metrik	46
5.4.7 OSPA-MT-Metrik.....	47
5.4.8 Intersection-over-Union	47
5.4.9 Evaluation von Trackingalgorithmen nach Morton et al.	48
5.5 Klassifikation	48

5.6	Gesamtmaß.....	52
5.7	Metrikenauswahl.....	53
6	Exemplarische Berechnung der Metriken.....	55
6.1	Aufzeichnen von realen Messdaten	55
6.1.1	Ermittlung der Winkelabweichung der Lidar-Messdaten.....	56
6.1.2	Ermittlung der Abweichung der Delta-Entfernung.....	58
6.2	Erstellen der Szenarien in VTD	61
6.3	Beschreibung der Szenarien.....	62
6.4	Anwendung der Metriken	65
6.4.1	Punktewolken.....	65
6.4.2	Tracking	82
6.4.3	Kombinierte Betrachtung der Verarbeitungsebenen „Punktewolken“ und „Tracking“	89
7	Diskussion der Ergebnisse	93
8	Fazit und Ausblick	96
	Anhang.....	99
A.1	Metriken für die Verarbeitungsebene Punktewolken.....	99
A.2	Zeitplan	103
	Literaturverzeichnis	104

Formelzeichen und Indizes

Lateinische Buchstaben:

Zeichen	Einheit	Bezeichnung
<i>A</i>	1	Menge
<i>A</i>	m ²	Fläche
<i>B</i>	1	Menge
<i>c</i>	m	Cut-Off-Parameter
<i>c</i>	$\frac{\text{m}}{\text{s}}$	Lichtgeschwindigkeit
<i>C</i>	1	Transportmatrix
<i>C</i>	1	Korrelationskoeffizient
<i>d</i>	m	Abstand
<i>D</i>	1	Entfernung
<i>f</i>	1	Histogramm-Klasse
<i>F</i>	1	Histogramm
<i>G</i>	1	Geometriefaktor
<i>h</i>	1	Histogramm-Klasse
<i>h</i>	m	Fahrzeughöhe
<i>H</i>	1	Histogramm
<i>J</i>	1	Jaccard-Index
<i>K</i>	$\text{kg} \frac{\text{m}^5}{\text{s}^5}$	Systemfaktor
<i>l</i>	m	Fahrzeuglänge
<i>m</i>	1	Anzahl der Trajektorien
<i>m</i>	1	Anzahl horizontaler Zellen
<i>n</i>	1	Anzahl der Trajektorien
<i>n</i>	1	Anzahl vertikaler Zellen
<i>n</i>	1	Brechungsindex
<i>N</i>	1	Partikelanzahl
<i>no</i>	1	Gesamtzahl der belegten Zellen
<i>nu</i>	1	Gesamtzahl der freien Zellen
<i>O</i>	1	Überlappungsfunktion
<i>p</i>	1	Präzision
<i>p</i>	1	Ordnung
<i>P</i>	W	Leistung
<i>r</i>	m	Erdradius
<i>r</i>	1	Trefferquote
<i>r</i>	m	Suchradius



Zeichen	Einheit	Bezeichnung
R	1	Rotationsmatrix
S	1	Wahrheitsmatrix
S	1	Teilmenge
t	s	Zeit
T	1	Teilmenge
T	1	Transmission
v	$\frac{\text{m}}{\text{s}}$	Geschwindigkeit
w	m	Fahrzeugbreite
x	m	Sensorfeste Koordinate
X	1	Teilmenge
y	m	Sensorfeste Koordinate
Y	1	Teilmenge
z	m	Sensorfeste Koordinate

Griechische Buchstaben:

Zeichen	Einheit	Bezeichnung
α	1	Auslöschungskoeffizient
α	1	Gewichtung des Benennungsfehlers
β	$\frac{1}{\text{m}^3}$	Rückstreukoeffizient
γ	1	Zuordnungsmenge
δ	°, rad	Orientierungswinkel der Fahrzeuge
\mathcal{A}	1	Gewichtung bei mehrfacher Zuweisung der Trajektorien
η	1	Wirkungsgrad
λ	°, rad	Längengrad
σ	m^2	Streuquerschnitt
σ	1	Standardabweichung
τ	s	Pulsdauer
Φ	°, rad	Winkelauflösung
ϕ	°, rad	Breitengrad
Ψ	°, rad	Gierwinkel
ω	1	Teilmenge
Ω	°, rad	Raumwinkel

Indizes:

Zeichen	Bezeichnung
abs	Absolut
B	Baron
B	Bhattacharyya
Br	Brechung
c	Cut-Off
diff	Differenz
H	Hausdorff
i, k, m, n	Laufindizes
p	Ordnung
P	Pearson
PP	Punktewolke-zu-Punktewolke
R	Real
rel	Relativ
S	Simuliert
scale	Skalierung
TOF	Time-of-Flight

Abkürzungsverzeichnis

ACC	Adaptive Cruise Control
ACPS	Automated Cyber Physical Systems
ADAS	Advanced Driver Assistance Systems
ADMA	Automotive Dynamic Motion Analyzer
AUC	Area under the ROC-Curve
BCC	Baron's cross-correlation coefficient
BMWi	Bundesministerium für Wirtschaft und Energie
CACC	Classification Accuracy
ECU	Electronic Control Unit
edm	Euclidean distance matrix
ENABLE-S3	European Initiative to Enable Validation for Highly Automated Safe and Secure Systems
FCR	Free Cells Ratio
FMEA	Fehlzustandsart- und -auswirkungsanalyse
FN	False Negative
FP	False Positive
FPR	False Positive Rate
FTA	Fault Tree Analysis
FZD	Fahrzeugtechnik Darmstadt
GOSPA	Generalized Optimal Subpattern Assignment
HA	Hinterachse
HAF	Hochautomatisierte Fahrfunktionen
HLF	High Level Fusion
IoU	Intersection over Union
k-NN	k-Nearest Neighbors
Lidar	Light Detection and Ranging
LLF	Low Level Fusion
MEMS	Microelectromechanical systems
MS	Map Score
OCACC	Overall Classification Accuracy
OCR	Occupied Cells Ratio
OE	Overall Error
OG	Occupancy Grid
OMAT	Optimal Mass Transfer
OPDF	Occupied Picture-Distance-Function
OSPA	Optimal Subpattern Assignment

OSPA-MT	Optimal Subpattern Assignment for Multiple Tracks
OSPA-T	Optimal Subpattern Assignment for Tracks
OSR	Over-segmentation Suppression Rate
PCC	Pearson's correlation coefficient
PDF	Picture-Distance-Function
pdf	Probability density function
PEGASUS	Projekt zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen zur Freigabe hochautomatisierter Fahrfunktionen
Radar	Radio Detection and Ranging
RANSAC	Random Sample Consensus
RBNN	Radially Bounded Nearest Neighbor
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
ROS	Robot Operating Systems
SAE	Society of Automotive Engineers
SG	Scan Grid
SIL	Software-in-the-Loop
SW	Software
TN	True Negative
TOF	Time-of-Flight
TP	True Positive
TPR	True Positive Rate
UPDF	Unoccupied Picture-Distance-Function
USR	Under-segmentation Suppression Rate
VA	Vorderachse
VTD	Virtual Test Drive

Abbildungsverzeichnis

Abbildung 1-1: Methodik des Vorgehens	3
Abbildung 2-1: Schematischer Lidar-Sensoraufbau	5
Abbildung 2-2: Pulsdiagramm (eigene Darstellung nach Gotzig).....	6
Abbildung 2-3: ibeo Lux 2010® vertikaler Öffnungswinkel mit vier Kanälen.....	8
Abbildung 2-4: ibeo Lux 2010® horizontaler Öffnungswinkel	8
Abbildung 2-5: ibeo Lux 2010® horizontale Winkelauflösung	9
Abbildung 2-6: Aufbau Lidar-Sensorsimulation	13
Abbildung 2-7: Visualisierung einer simulierten Punktwolke mit zwei Fahrzeugen	14
Abbildung 2-8: Adaption des Suchradius bei der Erweiterung des RBNN-Algorithmus	17
Abbildung 2-9: Struktur eines Trackingalgorithmus	19
Abbildung 2-10: Objekterkennung mit VoxelNet	20
Abbildung 4-1: Allgemeine Validierungsmethode nach Oberkamp und Trucano	29
Abbildung 4-2: Validierung von Sensormodellen nach Roth et al.	30
Abbildung 4-3: Validierung von Sensormodellen nach Schärmann et al.	31
Abbildung 4-4: Tool Chain zur Bewertung der Lidar-Sensorsimulation	32
Abbildung 5-1: Vergleich der Segmentierung	40
Abbildung 5-2: Sechs Szenarien zur Verdeutlichung der Stärken und Schwächen der Metriken.....	43
Abbildung 5-3: Schnittmenge und Vereinigungsmenge bei Objekten	48
Abbildung 5-4: ROC-Graph.....	52
Abbildung 6-1: Relativposition und -orientierung des ADMA	56
Abbildung 6-2: Vergleich Position des Corner Reflectors	57
Abbildung 6-3: Vergleich ADMA-Delta-Distanzen mit manuell berechneten Delta-Distanzen	60
Abbildung 6-4: Aufbau des Szenarios BMW „i3 statisch Heck/Front/Seite“ mit Abständen.....	63
Abbildung 6-5: Aufbau des Szenarios „Stauende 2 Fahrzeuge“ mit Abständen und Orientierung ..	63
Abbildung 6-6: Aufbau des Szenarios „parkende Autos“, alle Maße in mm	64
Abbildung 6-7: Szenario „Stauende dynamisch“, Screenshot aus VTD	65
Abbildung 6-8: Vergleich simulierte Punktwolke mit und ohne Bodenreflexionen.....	66
Abbildung 6-9: Beispielhaftes Occupancy Grid einer simulierten Punktwolke	68
Abbildung 6-10: Erstellung der verschiedenen OG-Varianten.....	69
Abbildung 6-11: Beispielhaftes gemitteltes Occupancy Grid einer realen Punktwolke	69
Abbildung 6-12: Vergleich der Varianten zur Erstellung der realen OGs.....	70
Abbildung 6-13: Einfluss der Anzahl der berücksichtigten Scans	71
Abbildung 6-14: 1-OE Ratio und BCC in Abhängigkeit des Abstandes	73
Abbildung 6-15: Variation des Suchradius der OPDF- und UPDF-Metrik.....	75
Abbildung 6-16: Variation der Zellauflösung, Szenario „Stauende 2 Fahrzeuge“, Abstand 10 m ...	78
Abbildung 6-17: OGs für den Vergleich der Zellauflösung, Szenario „Stauende 2 Fahrzeuge“	79
Abbildung 6-18: Variation der Zellauflösung und des Abstandes für die BCC-Metrik.....	80
Abbildung 6-19: OG-Metriken für den Vergleich mit und ohne BMW i3	81
Abbildung 6-20: Punktwolken-Metriken für den Vergleich mit und ohne BMW i3.....	81
Abbildung 6-21: Methodik zum Vergleich der Tracking-Ergebnisse	82
Abbildung 6-22: Reale und simulierte Trajektorien des Szenarios „Stauende dynamisch“	83
Abbildung 6-23: Trajektorien und OSPA-Metriken für das Szenario „Stauende dynamisch“	84

Abbildung 6-24: Intersection over Union in Abhängigkeit des Fahrzeugabstandes und Boxplots ...	86
Abbildung 6-25: Trajektorien und OSPA-Metriken des Szenarios „i3 Schrittgeschwindigkeit“	87
Abbildung 6-26: IoU-Metrik und Boxplots der Fahrzeuglänge und -breite	87
Abbildung 6-27: Methodik zur kombinierten Betrachtung der Verarbeitungsebenen	89
Abbildung 6-28: Tracking-Metriken für das Szenario „Stauende dynamisch“	91
Abbildung 6-29: Punktwolken-Metriken für das Szenario „Stauende dynamisch“	91

Tabellenverzeichnis

Tabelle 2-1: Einbauposition und -orientierung ibeo Lux 2010® im Versuchsfahrzeug.....	9
Tabelle 2-2: Merkmale nach Stäcker	21
Tabelle 3-1: FMEA zur Lidar-Sensorsimulation	24
Tabelle 5-1: Hit/Miss-Vergleich nach Browning et al.....	37
Tabelle 5-2: Übersicht Metriken für Punktwolken	38
Tabelle 5-3: Wahrheitsmatrix	49
Tabelle 5-4: Beispielhafte Wahrheitsmatrix mit Berechnung von CACC und OCACC	50
Tabelle 6-1: Vergleich gemessene und tatsächliche Position des Corner Reflectors	57
Tabelle 6-2: Überprüfung der ADMA-Distanz, alle Werte in mm.....	58
Tabelle 6-3: Vergleich der ADMA-Delta-Distanzen mit manuell berechneten Delta-Distanzen	61
Tabelle 6-4: Fahrzeugäquivalente in VTD.....	61
Tabelle 6-5: Metriken für das Szenario „Stauende 2 Fahrzeuge“ bei verschiedenen Abständen	77
Tabelle 6-6: RMSE für das Szenario „Stauende dynamisch“	85
Tabelle 6-7: Korrelationskoeffizienten zwischen den Punktwolken- und Tracking-Metriken.....	92
Tabelle A-1: Metriken für das Szenario „i3 statisch Heck“ bei verschiedenen Abständen.....	99
Tabelle A-2: Metriken für das Szenario „i3 statisch Front“ bei verschiedenen Abständen	100
Tabelle A-3: Metriken für das Szenario „i3 statisch Seite“ bei verschiedenen Abständen	101
Tabelle A-4: Metriken für das Szenario „parkende Autos“	102
Tabelle A-5: Zeitplan Masterthesis 699/18.....	103

1 Einleitung

Ein Trend in der Automobilindustrie ist die Entwicklung von Fahrzeugen mit zunehmend steigenden Automatisierungsstufen. Weit verbreitet ist heutzutage bereits die Autonomiestufe zwei, die nach SAE J3016¹ teilautomatisiertes Fahren mit Fahrfunktionen wie den Spurhalte- und Stauassistenten ermöglicht. Der Entwicklungsschwerpunkt von hoch- und vollautomatisierten Fahrzeugen liegt daher bei Assistenzsystemen ab der Autonomiestufe drei (SAE Level 3), die aufgrund der höheren Automatisierung und damit einhergehenden Komplexität neue Konzepte, Methoden und Qualitätsstandards zur Absicherung der Fahrfunktionen erfordern. Dies wird gerade nach dem tödlichen Unfall eines selbstfahrenden Autos des Fahrdienstvermittlers Uber im März 2018 deutlich, bei dem eine die Straße überquerende Fußgängerin von einem Versuchsfahrzeug erfasst wurde.²

1.1 Motivation

Die Absicherung des automatisierten Fahrens ist ein zentrales Thema des durch das Bundesministerium für Wirtschaft und Energie (BMWi) geförderten „Projekts zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen zur Freigabe hochautomatisierter Fahrfunktionen“ (PEGASUS³). Dieses Verbundprojekt, bei dem 17 Projektpartner aus Industrie und Wissenschaft zusammenwirken, hat sich zum Ziel gesetzt, ein einheitliches Vorgehen zum Testen und Erproben automatisierter Fahrfunktionen sowohl bei Simulationen als auch bei realen Fahrversuchen zu definieren. Dadurch soll die Einführung des automatisierten Fahrens in Serienanwendungen beschleunigt werden.

Das Fachgebiet Fahrzeugtechnik der TU Darmstadt (FZD) übernimmt im Rahmen von PEGASUS als Forschungseinrichtung Aufgaben in drei von vier Teilprojekten, unter anderem im Teilprojekt „Testen“. Das Ziel ist unter anderem das Entwickeln und die Validierung von Sensormodellen für aktive Sensoren wie Lidar-, Radar- und Ultraschallsensoren. Mithilfe dieser Modelle ergibt sich die Möglichkeit der Steigerung der Effizienz und der Wirtschaftlichkeit bei der Freigabe hochautomatisierter Fahrfunktionen im Gegensatz zu traditionellen Fahrversuchen.

Ein weiteres Projekt mit Beteiligung von FZD ist das ENABLE-S3-Projekt⁴ („European Initiative to Enable Validation for Highly Automated Safe and Secure Systems“), das von einem EU-weiten Konsortium mit 68 Projektpartnern aus sechs unterschiedlichen Bereichen (Automobil-, Luftfahrt-, Schienen-, See-, Gesundheits- und Landwirtschaftssektor) durchgeführt wird. Ziel des ENABLE-S3-Projektes ist das Ersetzen des bisher kostenintensiven Verifikations- und Validierungsprozesses durch neue Methoden, um die Voraussetzung für eine Kommerzialisierung von automatisierten cyber-physischen Systemen (ACPS) zu schaffen. Ermöglicht werden soll dies durch eine Kombination von

¹ Vgl. SAE International: J3016 (2018).

² Vgl. dpa: Uber-Unfall (2018).

³ Vgl. Bundesministerium für Wirtschaft und Energie: Homepage Forschungsprojekt PEGASUS (2016).

⁴ Vgl. ECSEL Joint Undertaking: Homepage ENABLE-S3 (2016).

Simulation und realen Erprobungen, die sich gegenseitig ergänzen und somit die jeweiligen Nachteile minimieren.

Die Sensormodelle, die für die beiden vorgestellten Projekte entwickelt werden und in den Simulationen zur Anwendung kommen, umfassen unterschiedliche aktive Sensoren. Das in dieser Masterthesis betrachtete Sensormodell beschränkt sich jedoch auf den Lidar-Sensor, daher werden im Folgenden der Radar- und der Ultraschallsensor nicht betrachtet. In vorausgegangenen Arbeiten wurde bei FZD ein Lidar-Sensormodell entwickelt, das kontinuierlich verbessert und mit neuen Funktionalitäten ausgestattet wird. Für die Validierung dieses Modells ist eine Bewertung auf den unterschiedlichen Verarbeitungsebenen erforderlich, die das Kernthema dieser Masterthesis darstellt.

1.2 Konkretisierung der Aufgabenstellung

Im Rahmen dieser Masterthesis soll eine Metrik für die Bewertung des vorhandenen Lidar-Sensormodells inklusive der simulierten Datenverarbeitung entwickelt werden. Für das Verständnis zur Funktionsweise des Lidar-Sensors sowie der nachfolgenden Verarbeitungskette soll eine umfassende Literaturrecherche durchgeführt werden. Hierfür werden neben der Fachliteratur die Datenblätter der im Versuchsfahrzeug verbauten Lidar-Sensoren ibeo LUX 2010® sowie die Dokumentation⁵ des Sensorherstellers herangezogen. Anschließend soll eine Bewertung der vorhandenen Verarbeitungskette erfolgen. Die Bewertung ist mit einem Verfahren zur Zuverlässigkeitsanalyse, zum Beispiel mittels FMEA oder FTA (Fault Tree Analysis), durchzuführen.

Im nächsten Schritt soll eine belastbare Literaturübersicht zu den Metriken zur Bewertung der unterschiedlichen Verarbeitungsebenen erstellt werden. Zu den Verarbeitungsebenen gehören die Ausgabe der Punktwolke, die Segmentierung, das Verfolgen der Segmente sowie das anschließende Klassifizieren dieser. Da die Literatur zu diesen Themen zum Großteil in englischer Sprache gehalten ist, wird im Folgenden teilweise auf die gleichbedeutenden Begriffe *Point Cloud*, *Segmentation*, *Tracking* und *Classification* zurückgegriffen. Die Erläuterung der einzelnen Ebenen erfolgt im Abschnitt 2.3.

Geeignete Metriken sollen für jede Verarbeitungsebene ausgewählt und gegenübergestellt werden. Ausgehend von diesen Metriken sind mögliche Wege aufzuzeigen, wie aus den einzelnen Metriken ein Gesamtmaß entwickelt werden kann. Dieses Gesamtmaß soll aus der Summe der ausgewählten Kriterien bestehen und zudem auf unterschiedliche Sensor-Charakteristiken anwendbar sein. Für die Bewertung des Gesamtmaßes sind die Kriterien Anwendbarkeit und Generalisierbarkeit von Bedeutung. Wenn kein Gesamtmaß entwickelt wird, ist dies zu begründen.

Unter Verwendung der ausgewählten Metriken für die einzelnen Verarbeitungsebenen sowie des möglichen entwickelten Gesamtmaßes ist das Lidar-Sensormodell zu bewerten. Hierzu werden exemplarische Berechnungen an einfachen Szenarien durchgeführt, beispielsweise die Auffahrt auf ein Stauende. Falls eine Metrik ein spezielles Szenario erfordert, so ist dieses zu definieren.

⁵ Vgl. Ibeo Automotive Systems GmbH: Operating Manual ibeo LUX 2010® (2014).

1.3 Methodik des Vorgehens

Anhand der Abbildung 1-1 wird in diesem Abschnitt die Methodik für die Bearbeitung der Masterthesis vorgestellt. Der erste Teil stellt die Literaturrecherche dar. Sie umfasst das Recherchieren von Informationen über den Lidar-Sensor, der Sensormodellierung, der Validierung von Sensormodellen und Metriken zur Bewertung von Sensormodellen. Die Ergebnisse der Literaturrecherche zu den Themen Lidar-Sensor und Sensormodellierung sind in den theoretischen Grundlagen in Kapitel 2 niedergeschrieben. Dieses Kapitel beinhaltet außerdem eine Erklärung zum Aufbau des Lidar-Sensormodells bei FZD.

Der zweite Teil der Masterthesis beginnt mit der Bewertung der vorhandenen Verarbeitungskette mittels einer FMEA oder einer FTA. Kapitel 3 widmet sich diesem Thema und der Fragestellung, ob eine FMEA oder FTA sinnvoll für die Bewertung einer Sensorsimulation verwendet werden kann. Im Kapitel 4 sind die Ergebnisse der Literaturrecherche über die Methodik zur Bewertung und Validierung eines Sensormodells zusammengefasst, des Weiteren wird die in dieser Masterthesis entwickelte Tool Chain zur Bewertung des Sensormodells vorgestellt. Im Anschluss werden in Kapitel 5 die Metriken vorgestellt, die für die Bewertung der Lidar-Sensorsimulation eingesetzt werden. Diese sind unterteilt in verschiedene Verarbeitungsebenen. Das Kapitel 6 befasst sich mit der exemplarischen Berechnung der Metriken für ausgewählte Szenarien. Hierfür werden zunächst beispielhafte Szenarien definiert und für diese die Metriken für die unterschiedlichen Verarbeitungsebenen berechnet. Im Kapitel 7 werden die erzielten Ergebnisse kritisch diskutiert. Abschließend folgt in Kapitel 8 das Fazit sowie der Ausblick. Alle Arbeitsschritte werden kontinuierlich dokumentiert. Im Anhang A.2 befindet sich der Zeitplan dieser Masterthesis mit einem Soll-Ist-Vergleich.

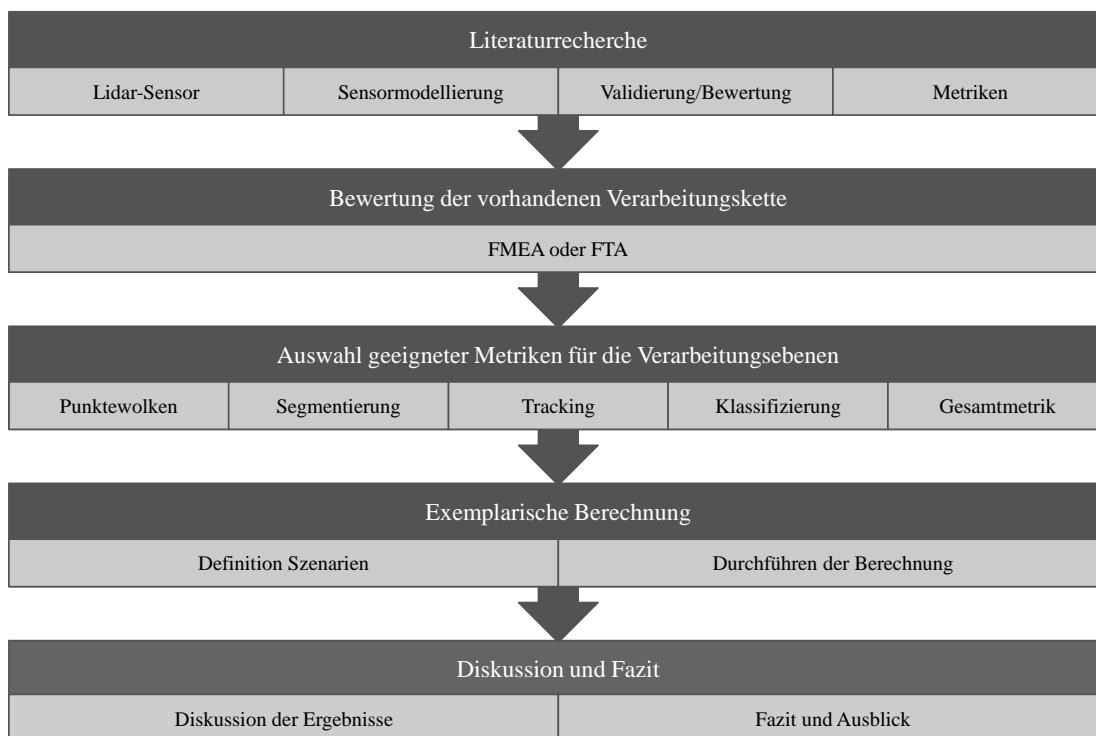


Abbildung 1-1: Methodik des Vorgehens

2 Theoretische Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen des Lidar-Sensors sowie der Stand der Technik der Lidar-Sensorsimulation erläutert. Im ersten Abschnitt wird der Lidar-Sensor näher betrachtet. Hierzu werden der Aufbau und das Funktionsprinzip des Lidar-Sensors sowie die allgemeine Lidar-Gleichung erklärt. Des Weiteren wird auf die Spezifikationen des in der vorliegenden Ausarbeitung verwendeten Lidar-Sensors ibeo Lux 2010® eingegangen. Im zweiten Abschnitt dieses Kapitels werden Ansätze zu Sensormodellen in der Simulation, unabhängig vom spezifischen Sensor, diskutiert und die Frage, wozu Sensormodelle benötigt werden, beantwortet. Das Kapitel wird abgeschlossen mit der Vorstellung des Lidar-Sensormodells, welches bei FZD verwendet wird.

2.1 Der Lidar-Sensor

Der Lidar-Sensor (Light Detection And Ranging) ermöglicht das Lokalisieren von Objekten im Raum und ermittelt zudem die Entfernung, Geschwindigkeit und Orientierung dieser Objekte.⁶ Das Messverfahren des Lidar-Sensors zählt, im Gegensatz zum elektromagnetischen Messverfahren des Radar-Sensors (Radio Detection And Ranging), zu den optischen Messverfahren. Ähnlich wie bei Radar-Sensoren basiert die Messung auf dem Emittieren von Strahlen, deren Wellenlänge bei Lidar-Sensoren jedoch im Bereich des sichtbaren Lichts liegt (in der Automobilanwendung: 850 nm bis 1 μm ⁷). Die ausgesendeten fokussierten Lichtpulse werden von Objekten im nahen Umfeld reflektiert und die zurückfallenden Lichtstrahlen durch die Empfangseinheit des Lidar-Sensors detektiert. Die Abstandsmessung erfolgt mittels anschließender Laufzeitmessung *Time-of-Flight* (TOF).

2.1.1 Aufbau und Funktionsprinzip des Lidar-Sensors

Prinzipiell sind die heutzutage erhältlichen Lidar-Sensoren ähnlich aufgebaut. Sie bestehen aus einem Sendezweig, der für die Strahlenerzeugung und -bündelung zuständig ist sowie aus einem Empfangszweig, der alle einfallenden Photonen aufnimmt und verarbeitet. Im oberen Teil der Abbildung 2-1, die den schematischen Aufbau eines Lidar-Sensors zeigt, ist der Sendezweig dargestellt. Dieser beinhaltet unter anderem die Laserquelle, die von einem Lichtpuls-Generator angesteuert wird sowie den Scanner, der beispielsweise aus einem die Strahlen ablenkenden rotierenden Spiegel besteht.

Es sei erwähnt, dass neben dem rotierenden Spiegel als Scanner noch weitere Ausführungen existieren, bei denen z. B. der Lidar-Sensor um seine eigene Achse rotiert (siehe Velodyne HDL-64E⁸). Die aktuellen Entwicklungen im Bereich der Festkörper-Lidar-Sensoren, die auf bewegliche Komponenten zur Strahlenausrichtung verzichten und stattdessen auf phasengesteuerten Arrays⁹ oder MEMS-Spiegeln¹⁰ (microelectromechanical systems) basieren, stellen Alternativen zu den konventionellen

⁶ Vgl. Kernhof, J. et al.: LiDAR-Sensorsystem (2018), S. 30.

⁷ Vgl. Gotzig, H.; Geduld, G.: LIDAR-Sensorik (2015), S. 320.

⁸ Vgl. Velodyne LiDAR Inc.: HDL-64E S3 (2018).

⁹ Vgl. Dubois, C.: Solid State LiDAR (2018).

¹⁰ Vgl. Sandner, T. et al.: MEMS Mirror Array (2008).

mechanischen Lidar-Sensoren dar und bieten Vorteile hinsichtlich des Bauraums, der Robustheit und der Kosten.¹¹

Die der Photodiode nachgeschaltete Verarbeitungskette im Empfangszweig (untere Hälfte in Abbildung 2-1), bestehend aus einem Verstärker und einem Analog-Digital-Wandler mit vorgelagerter Sample-and-Hold-Schaltung, dient der Digitalisierung des analogen Signals für die anschließende Auswertung. Ein Mikrocontroller übernimmt das Timing der Ansteuerung des Sende- und Empfangszweigs.

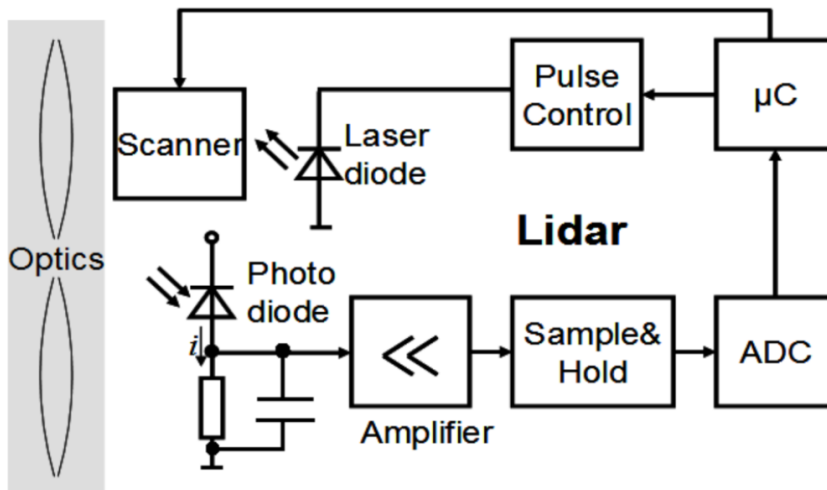


Abbildung 2-1: Schematischer Lidar-Sensoraufbau¹²

Das Messprinzip des Lidar-Sensors beruht auf der Messung der Laufzeit des Lichtpulses. Die Dauer des emittierten Pulses liegt zwischen 4 und 30 ns^{13a}, wobei eine möglichst kurze Pulsdauer angestrebt wird, da diese die räumliche Auflösung beschränkt. Eine Pulsdauer von 10 ns entspricht einer räumlichen Pulslänge von 3 m und kann zu einer Überlagerung der reflektierten Pulse bei Objekten führen, die in geringem Abstand hintereinander angeordnet sind.¹⁴ Wenn sich mehrere Objekte in genügend großem Abstand zueinander innerhalb eines Lichtstrahles befinden, so reflektieren diese jeweils einen Teil des emittierten Lichtpulses zum Empfänger zurück. Diese zeitlich verzögerten Pulsantworten werden den einzelnen Objekten zugeordnet, wenn das System mit einer Mehrzielfähigkeit ausgestattet ist.^{13b}

Grundsätzlich hat die Pulsantwort bei kurzen Lichtpulsen die Form einer Gauß-Verteilung, siehe Abbildung 2-2. Die Laufzeit zwischen dem Aussenden des Pulses und dem Empfangen der Pulsantwort wird als Time-of-Flight t_{TOF} bezeichnet. Mittels ermittelter Laufzeit und der Lichtgeschwindigkeit im

¹¹ Vgl. Ross, P. E.: Solid-State Lidar (2018).

¹² Winner, H.: Vorlesungsunterlagen Mechatronik und Assistenzsysteme im Automobil (2017), S. 8.

¹³ Vgl. Gotzig, H.; Geduld, G.: LIDAR-Sensorik (2015), a: S. 321; b: S. 319.

¹⁴ Vgl. Kernhof, J. et al.: LiDAR-Sensorsystem (2018), S. 42.

Vakuum $c_0 = 299.792.458 \frac{\text{m}}{\text{s}}$ ergibt sich für den Abstand d zwischen Lidar-Sensor und reflektierendem Objekt

$$d = \frac{c_0}{n_{\text{Br}}} \cdot \frac{t_{\text{TOF}}}{2}. \quad (2-1)$$

In Gleichung (2-1) wird der Brechungsindex n_{Br} des Mediums berücksichtigt, der im Allgemeinen abhängig von der Wellenlänge ist und das Verhältnis zwischen der Lichtgeschwindigkeit im Vakuum zur Ausbreitungsgeschwindigkeit des Lichts im Medium beschreibt. Außerdem wird bei der Abstandsermittlung beachtet, dass der Lichtpuls den Weg vom Lidar-Sensor zum Objekt zweimal durchläuft.

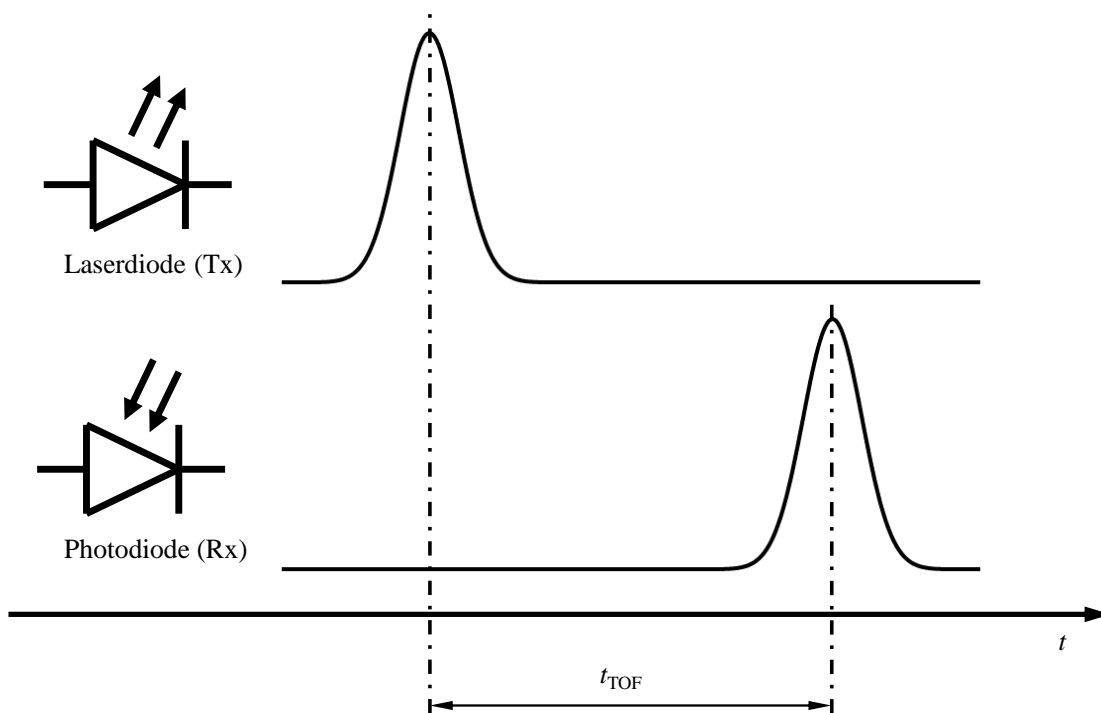


Abbildung 2-2: Pulsdiagramm (eigene Darstellung nach Gotzig¹⁵)

2.1.2 Beschreibung der Lidar-Gleichung

Die empfangene Lichtintensität P in einem Lidar-Sensor berechnet sich mittels der allgemeinen Lidar-Gleichung nach Wandinger^{16a} zu

$$P(d) = K G(d) \beta(d) T(d), \quad (2-2)$$

wobei die Intensität eine Funktion des Messabstandes d ist, der zwischen dem reflektierenden Objekt und der Empfangseinheit vorliegt. Die vier Faktoren werden als Systemfaktor K , Geometriefak-

¹⁵ Vgl. Gotzig, H.; Geduld, G.: LIDAR-Sensorik (2015), S. 319.

¹⁶ Vgl. Wandinger, U.: Introduction to Lidar (2005), a: -; b: S. 6; c: S. 7.

tor $G(d)$, Rückstreukoeffizient $\beta(d)$ und Transmissionsterm $T(d)$ bezeichnet. Während die ersten beiden Faktoren direkt durch den Aufbau des Lidar-Sensors bestimmt sind, hängen der Rückstreukoeffizient und der Transmissionsterm vom atmosphärischen Umfeld ab.^{16b}

Bei der Entwicklung eines Lidar-Sensors ist der Systemfaktor K der primäre Designparameter.^{16c} Dieser ist definiert zu

$$K = P_0 \frac{c\tau}{2} A\eta \quad (2-3)$$

und kann beispielsweise durch Variation der Pulsenergie $E_0 = P_0\tau$ verändert werden, wobei P_0 die durchschnittliche Leistung eines einzelnen Laserpulses und τ die zeitliche Länge dieses Pulses darstellt. Eine weitere Möglichkeit zur Änderung des Systemfaktors besteht in der Vergrößerung oder Verkleinerung der Fläche A der Empfangsoptik oder in der Optimierung des Systemwirkungsgrades η .

Der Geometriefaktor $G(d)$ ist durch die Überlappungsfunktion des Empfängers $O(d)$ und dem Term d^{-2} beschrieben. Die Überlappungsfunktion nimmt bei großen Entfernungen (mehrere Kilometer, vgl. Wandinger^{16a}) den Wert Eins an und nähert sich im Nahbereich des Lidar-Sensors dem Wert Null an, da im Nahbereich nicht der gesamte ausgesendete Laserpuls auf dem Empfangsdetektor abgebildet wird. Für eine ausführliche Herleitung der Überlappungsfunktion sei auf Wandinger^{16a} hingewiesen.

$$G(d) = \frac{O(d)}{d^2} \quad (2-4)$$

Der Rückstreukoeffizient

$$\beta(d) = \sum_i N_i(d) \frac{d\sigma_i}{d\Omega} \quad (2-5)$$

ist ein Sonderfall des allgemeinen Rückstreukoeffizienten für einen Rückstreuwinkel von $\theta = 180^\circ$. Er beschreibt, wie viel Licht von allen Partikeln N des Typs i im Streuquerschnitt $d\sigma/d\Omega$ zum Empfänger reflektiert wird.

Der letzte Term der Lidar-Gleichung ist der Transmissionsterm $T(d)$, in dem die Lichtauslöschung durch Streuung und Absorption, die auf dem Hin- und Rückweg zwischen Lidar-Sensor und Objekt entsteht, Berücksichtigung findet. Mit dem Auslöschungskoeffizienten $\alpha(r)$ und dem Integral über den Weg r zum Abstand d ist der Transmissionsterm definiert zu

$$T(d) = \exp \left[-2 \int_0^d \alpha(r) dr \right]. \quad (2-6)$$

Für die allgemeine Lidar-Gleichung (2-2) folgt durch Einsetzen der Gleichungen (2-3) bis (2-6)

$$P(d) = P_0 \frac{c\tau}{2} A\eta \frac{O(d)}{d^2} \sum_i \left(N_i(d) \frac{d\sigma_i}{d\Omega} \right) \exp \left[-2 \int_0^d \alpha(r) dr \right]. \quad (2-7)$$

2.1.3 Spezifikationen des ibeo Lux 2010®

Der im Versuchsfahrzeug sowie in der Simulation zum Einsatz kommende Lidar-Sensor ist der ibeo Lux 2010® Laserscanner. Die folgenden technischen Daten des Lidar-Sensors sind aus der Gebrauchsanweisung^{17a} entnommen. Der Lidar-Sensor verfügt über vier unabhängige Empfänger, die vier vertikal angeordnete Kanäle mit einem Öffnungswinkel von jeweils $0,8^\circ$ ermöglichen (siehe Abbildung 2-3). Insgesamt besitzt der ibeo Lux somit einen vertikalen Öffnungswinkel von $3,2^\circ$. Bei vier aktiven Kanälen beträgt der horizontale Öffnungswinkel 85° (siehe Abbildung 2-4, Bereich 1), eine Erweiterung des Öffnungswinkels auf bis zu 110° ist möglich, wobei der erweiterte Bereich in diesem Fall nur von zwei Kanälen abgedeckt wird (siehe Abbildung 2-4, Bereich 2).

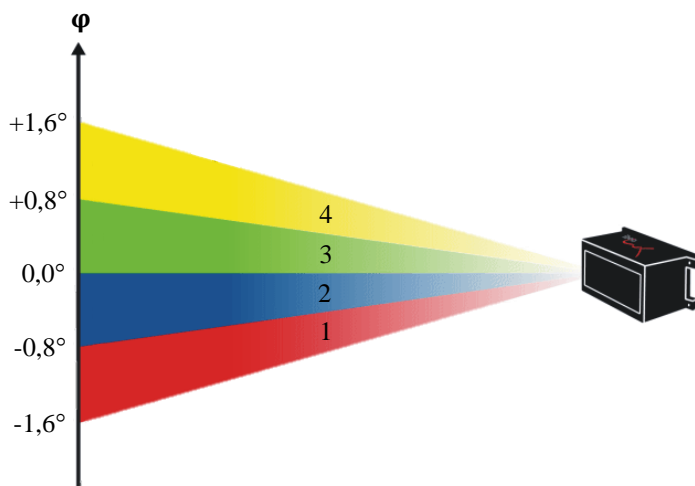


Abbildung 2-3: ibeo Lux 2010® vertikaler Öffnungswinkel mit vier Kanälen^{17b}

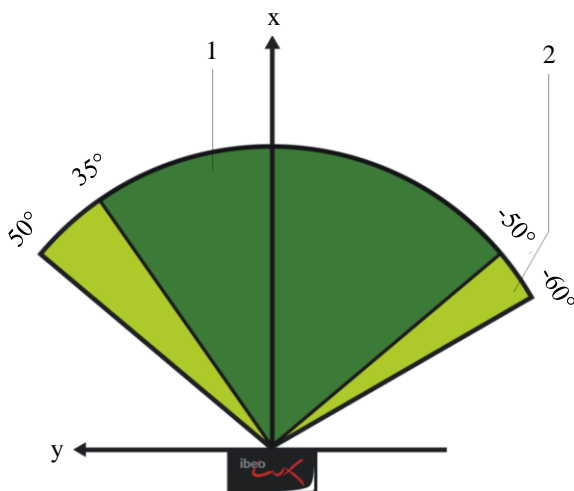


Abbildung 2-4: ibeo Lux 2010® horizontaler Öffnungswinkel^{17c}

¹⁷ Vgl. Ibeo Automotive Systems GmbH: Operating Manual ibeo LUX 2010® (2014), a: -; b: S. 3-5; c: S. 3-12; d: S. 3-13.

Die horizontale Winkelauflösung (Abbildung 2-5) zwischen den einzelnen Strahlen ist von der Scanfrequenz abhängig und nimmt Werte zwischen $0,125^\circ$ und $0,5^\circ$ an. Dabei werden immer zwei Kanäle (Kanalpaar 1 und 2 (unten) oder 3 und 4 (oben), siehe Abbildung 2-5) gleichzeitig erfasst, sodass der Winkelabstand zwischen zwei Strahlen innerhalb eines Kanals dem doppelten Winkelabstand zwischen zwei Strahlen unterschiedlicher Kanalpaare entspricht. Es ist auch möglich, verschiedenen Winkelbereichen unterschiedliche Winkelauflösungen zuzuweisen. Die Entfernungsauflösung des Lidar-Sensors beträgt 0,04 m, die Reichweite ist mit maximal 200 m in 0° Richtung angegeben.

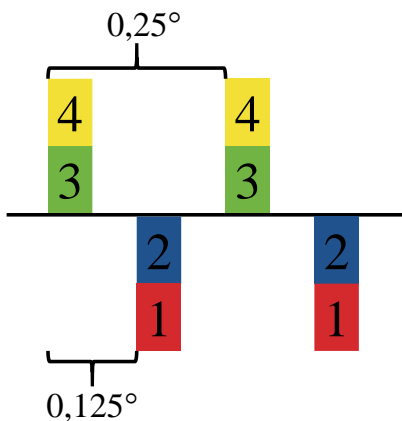


Abbildung 2-5: ibeo Lux 2010® horizontale Winkelauflösung^{17d}

Im realen Versuchsfahrzeug von FZD sind zwei ibeo Lux 2010® in der vorderen Stoßstange links und rechts verbaut, die jeweils um $17,9^\circ$ von der Fahrzeugmitte nach außen zeigend gedreht sind. Der horizontale Öffnungswinkel beider Lidar-Sensoren ist auf 110° konfiguriert (Vergleich Abbildung 2-4, Bereich 2) und ermöglicht somit eine vollständige Abdeckung bereits ab wenigen Metern vor dem Versuchsfahrzeug. Zusätzlich wird ein breiter Bereich auf beiden Seiten vor dem Fahrzeug erfasst. Die genauen Einbaupositionen sowie -orientierungen der Lidar-Sensoren sind in der Tabelle 2-1 aufgelistet. Die Angaben beziehen sich auf ein kartesisches Koordinatensystem, dessen Ursprung auf der Fahrbahnebene in der Mitte der Hinterachse des Fahrzeuges liegt, wobei die x-Achse für positive Werte in Fahrtrichtung und die y-Achse für positive Werte quer zur Fahrtrichtung nach links zeigt. Da die ibeo Lux 2010® Sensoren auch in der Simulation modelliert sind, gleichen die Einbaupositionen und -orientierungen der Sensoren in der Simulation denen in der Tabelle 2-1.

Tabelle 2-1: Einbauposition und -orientierung ibeo Lux 2010® im Versuchsfahrzeug

	Versatz X	Versatz Y	Versatz Z	Neigungswinkel	Wankwinkel	Gierwinkel
Lidar 1	3,47 m	0,63 m	0,36 m	0°	0°	$17,906^\circ$
Lidar 2	3,49 m	-0,58 m	0,36 m	0°	0°	$-17,906^\circ$

Der Sensor bestimmt die Distanzen der reflektierenden Objekte mittels Laufzeitmessung TOF, wobei pro ausgesendetem Lichtimpuls bis zu drei Echos detektiert werden. Die auf diese Weise erzeugte Punktwolke wird in einer anschließenden integrierten Auswertung verarbeitet. Wenn mehrere ibeo

Lux 2010® in einem Fahrzeug verbaut sind, so werden die Punktwolken, die in Polarkoordinaten vorliegen, von einer Electronic Control Unit (ECU) fusioniert und anschließend als Punktwolke in einem kartesischen Koordinatensystem ausgegeben. Der erste Schritt der Verarbeitungskette ist das Detektieren von Bodenreflexionen, die als solche markiert werden. Im nächsten Schritt wird die Punktwolke segmentiert, indem Gruppierungen von Punkten zu einzelnen Segmenten zusammengefasst werden. Der dritte Verarbeitungsschritt ist das sogenannte Tracking, also das Verfolgen von Segmenten über einen längeren Zeitraum und die Berechnung der Geschwindigkeit der getrackten Objekte unter Berücksichtigung der Ego-Fahrzeuggeschwindigkeit. In einem letzten Auswerteschritt kommt ein Klassifikator zum Einsatz, der auf Basis der Objekthistorie und -eigenschaften eine Einteilung in verschiedene Klassen (Pkw, Zweirad, Lkw, Fußgänger, unbekannt groß und unbekannt klein) vornimmt.¹⁸

2.2 Sensoren in der Simulation

Die Absicherung und die Freigabe des automatisierten Fahrens erfordert viele Testkilometer (Abschätzung siehe Wachenfeld et al.¹⁹), um beispielsweise seltene Fehler in sicherheitskritischen Szenarien zu entdecken. Eine Möglichkeit neben den realen Fahrversuchen ist das Einsetzen von virtuellen Fahrversuchen. Hierbei wird die Methode Software-in-the-Loop (SIL) verwendet, bei der eine Software oder ein Algorithmus in eine virtuelle Testumgebung eingebracht wird und die Funktionen der zu testenden Software überprüft werden. Auf diese Weise ist es möglich, komplexe Software virtuell zu erproben, bevor sie in der realen Welt getestet wird.²⁰ Ein Vorteil der SIL-Methode ist, dass die Softwarefunktionen ohne die Hardware, die die entsprechenden Eingangsdaten für die Software liefert oder die Ausgangsdaten der Software empfängt, bereits getestet werden und somit Entwicklungszeit eingespart wird, da die Zeit bis zum Testen der Software in realen Fahrzeugprototypen überbrückt wird.^{21a}

Im Kontext des autonomen Fahrens werden *Advanced Driver Assistance Systems* (ADAS) verwendet, also beispielsweise das adaptive Geschwindigkeitsregelsystem ACC (*Adaptive Cruise Control*). Bei autonomen Fahrzeugen zählt der ADAS-Bereich zu den sicherheitskritischen Systemen, die es abzusichern gilt und deren Zuverlässigkeit von großer Bedeutung ist. Die meisten Eingangsdaten der ADAS-Funktionen stammen von Sensoren, die das Umfeld des Fahrzeuges wahrnehmen.²² Um die ADAS-Funktionen und -Robustheit virtuell zu testen, sind demnach auch virtuelle Sensoren notwendig, die die Funktionen eines realen Sensors in einer virtuellen Umgebung abbilden. Diese virtuellen Fahrversuche haben gegenüber den realen Fahrversuchen den Vorteil, dass die Ergebnisse in der Simulation reproduzierbar sind und dadurch beispielsweise verschiedene Softwareversionen in gleichen Szenarien miteinander verglichen werden können.

¹⁸ Vgl. Ibeo Automotive Systems GmbH: Operating Manual ibeo LUX 2010® (2014), S. 4-4–4-8.

¹⁹ Vgl. Wachenfeld, W.; Winner, H.: Die Freigabe des autonomen Fahrens (2015), S. 454–458.

²⁰ Vgl. Ilic, V. et al.: ADAS Modules in Virtual Environments (2018), S. 88.

²¹ Vgl. Hakuli, S.; Krug, M.: Virtuelle Integration (2015), a: S. 126; b: S. 137.

²² Vgl. Hirsenkorn, N. et al.: Virtual sensor models for real-time applications (2016), S. 31.

Ziel der Sensorsimulation ist es, dass der Ausgang des Sensormodells und damit der Eingang einer ADAS-Funktion nicht von dem Ausgang eines vergleichbaren realen Sensors zu unterscheiden ist.²³ Wichtig bei der Sensormodellierung ist, dass auf der einen Seite die Funktionen des Sensors realitätsgetreu in der virtuellen Simulationsumgebung abgebildet werden, auf der anderen Seite jedoch der Simulations- und Rechenaufwand nicht zu groß werden darf. Diese Optimierungsaufgabe zwischen der Detaillierung und dem Rechenzeitbedarf ist speziell für Echtzeitanwendungen eine Herausforderung.^{21b}

Nach Holder et al. werden Sensormodelle in die drei Kategorien ideale Modelle, phänomenologische Modelle und physikalische Modelle eingeteilt.²⁴ Bei idealen Sensormodellen werden keine Sensorfehler modelliert, sodass die Ausgabe des Sensormodells eine ideale Objektliste darstellt, die alle Objekte enthält, die von der virtuellen Simulationsumgebung dargestellt werden.

Phänomenologische Sensormodelle berücksichtigen zusätzlich Sensoreigenschaften wie z. B. das Sichtfeld, die limitierte Auflösung oder Sensorrauschen.²⁴ Hirsenkorn et al. implementieren in einem Radar-Sensormodell mit einem nicht-parametrischem Ansatz die Modellierung einer Wahrscheinlichkeitsdichtefunktion, mit deren Hilfe ein beliebiger diskreter Ausgang der Objektliste des Sensors – beispielsweise die Entfernung des Objekts – entsprechend der Wahrscheinlichkeitsdichtefunktion die Verteilung dieser annimmt.²⁵ Für die Implementierung wird vorgeschlagen, dass – für den Fall der Entfernung zu einem Objekt – mehrere Entfernungsmessungen durchgeführt werden und hierbei die Referenzentfernung (*Ground Truth*) sowie die Entfernung als Ausgabe des realen Sensors gemessen werden. Beide Signale werden jeweils mit einer Gauß-Verteilung beaufschlagt und die Ground-Truth-Distanzen den gemessenen Sensordistanzen zugeordnet. Nach diesem Verfahren existiert für jede Distanz im gemessenen Bereich eine Wahrscheinlichkeitsdichtefunktion für die Sensorausgabe. Diese Wahrscheinlichkeitsdichtefunktion wird in der Simulation übernommen.²⁵

Die physikalischen Modelle stellen die dritte Kategorie der Sensormodelle dar. Bei der Umsetzung dieser wird versucht, die Rohdaten des realen Sensors mithilfe des zugrunde liegenden physikalischen Prinzips nachzustellen. Gschwandtner et al. zeigen als mögliches Realisierungsbeispiel ein Sensormodell, das auf Raytracing basiert. Wie bei einem realen Lidar-Sensor wird beim Raytracing jeder Lichtstrahl einzeln simuliert. Das Sensormodell von Gschwandtner et al. ermöglicht es zudem, die Materialeigenschaften der Oberflächen, die von den Lichtstrahlen getroffen werden, auszulesen.²⁶

Eine Herausforderung bei der Modellierung eines Lidar-Sensors ist die Strahlaufweitung. Mit zunehmender Entfernung vom Lidar-Sensor vergrößert sich der Querschnitt eines Strahls und somit die Fläche, in der ein potentiell Objekt liegen kann. Außerdem können mehrere Objekte innerhalb eines Strahls zu mehreren Reflexionen führen. Um dies zu berücksichtigen muss die entfernungsabhängige Querschnittsfläche der Strahlen modelliert werden.

²³ Vgl. Hanke, T. et al.: Generic Architecture for Simulation of ADAS Sensors (2015).

²⁴ Vgl. Holder, M. et al.: Requirements for a Lidar Sensor Model (2018), S. 2.

²⁵ Vgl. Hirsenkorn, N. et al.: A Non-Parametric Approach for Modeling Sensor Behavior (2015).

²⁶ Vgl. Gschwandtner, M. et al.: BlenSor: Blender Sensor Simulation Toolbox (2011), S. 201.

Zusätzlich ist bei der Modellierung eines Lidar-Sensors zu beachten, dass zeitliche Effekte Berücksichtigung finden. Bei einem Lidar-Sensor, der die Strahlen mit einem rotierenden Scanner ablenkt, werden die Punkte eines Scans zeitlich nacheinander erfasst. Dies kann zu Rolling Shutter und Motion Blur führen, wenn sich ein Objekt im Sichtbereich des Lidar-Sensors bzw. das Ego-Fahrzeug bewegt. Rosenberger et al. erklären diese Effekte und zeigen einen Ansatz, wie diese in einem Lidar-Sensormodell implementiert werden können.²⁷

²⁷ Vgl. Rosenberger, P. et al.: Analysis of Real World Sensor Behavior (2018), S. 615.

2.3 Lidar-Sensorsimulation bei FZD

In diesem Abschnitt wird der Aufbau des Lidar-Sensormodells, welches bei FZD entwickelt wird, erläutert. Neben dem Lidar-Sensormodell existieren bei FZD noch weitere Sensormodelle für Radar und Ultraschall, deren Funktionsweise im Folgenden jedoch nicht erklärt wird.

Der grundsätzliche Aufbau des Lidar-Sensormodells wird anhand der Abbildung 2-6 erklärt. Die gesamte Lidar-Sensorsimulation ist in zwei Teile untergliedert. Eine virtuelle Simulationsumgebung bildet das erste Segment. Hierbei wird auf das Simulationstool *Virtual Test Drive* (VTD) von Vires²⁸ zurückgegriffen. Dieses bietet die Möglichkeit, eine gesamte Simulationsumgebung aus einzelnen Bausteinen aufzubauen und in dieser Umgebung das Versuchsfahrzeug (Ego-Fahrzeug) sowie weitere Fahrzeuge auf definierbaren Pfaden zu bewegen. Das Ego-Fahrzeug kann dabei mit unterschiedlichen Sensoren ausgestattet werden. Im Falle der Sensorsimulation von FZD sind am Ego-Fahrzeug zwei virtuelle Lidar-Sensoren angebracht, die jeweils ein Sensormodell des ibeo Lux 2010® Laserscanners darstellen und deren Einbaupositionen und -orientierungen mit denen des realen Referenzfahrzeugs übereinstimmen (siehe Tabelle 2-1).

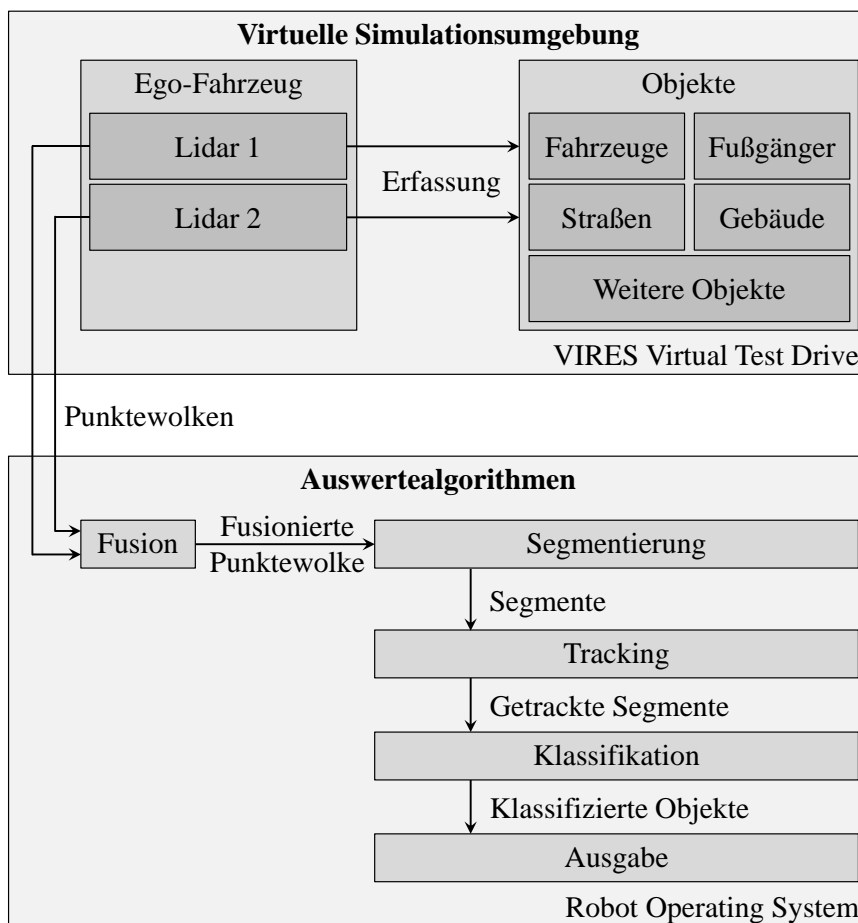


Abbildung 2-6: Aufbau Lidar-Sensorsimulation

²⁸ Vgl. VIRE Simulationstechnologie GmbH: Virtual Test Drive (2018).

Das Sensormodell basiert auf der Methode des Raytracing bzw. Raylaunching (vgl. Abschnitt 2.2). Alle ausgesendeten Lichtstrahlen werden von den virtuellen Objekten in VTD reflektiert und alle Reflexionspunkte von den beiden Sensoren jeweils als 3D-Punktwolke ausgegeben. Mithilfe des *Robot Operating Systems* (ROS) ist es möglich, die Punktwolken aller Zeitschritte der Simulation in einer *bag*-Datei zu speichern. Mit dem 3D-Visualisierungstool *rviz* von ROS besteht die Möglichkeit, die Punktwolken während der Simulation darzustellen. In Abbildung 2-7 ist eine simulierte Punktwolke gezeigt, die die Reflexionen von zwei Fahrzeugen, die in Längsrichtung vor dem Ego-Fahrzeug stehen, darstellt. Bei dem linken Fahrzeug sind Reflexionspunkte an der Seite des Fahrzeuges sichtbar, da dieses Fahrzeug einen lateralen Versatz zum Ego-Fahrzeug aufweist (Ego-Fahrzeug nicht in der Abbildung dargestellt).

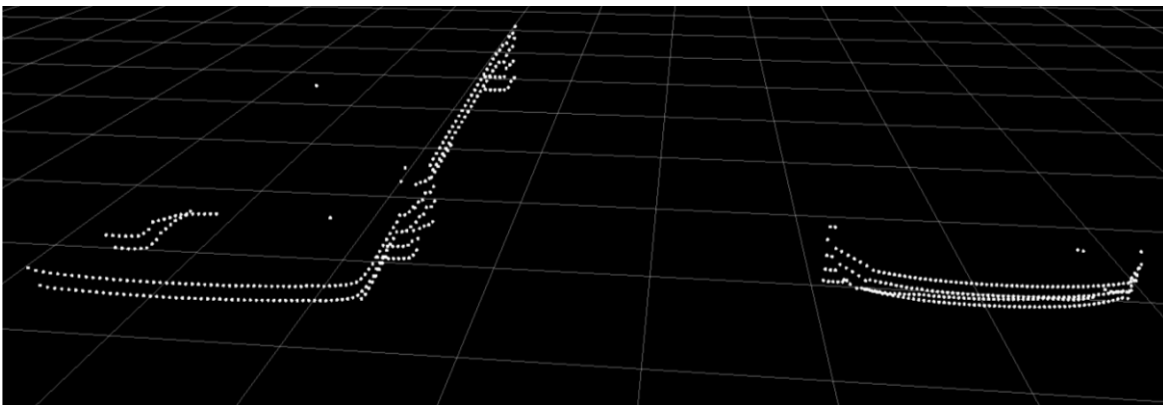


Abbildung 2-7: Visualisierung einer simulierten Punktwolke mit zwei Fahrzeugen

Der zweite Teil der Lidar-Sensorsimulation (vgl. Abbildung 2-6) besteht aus den Auswerteargorithmen, die jeweils als eigenes Modul in ROS einbettet sind und in den folgenden Abschnitten beschrieben werden. Die mittels VTD generierten Punktwolken der beiden Lidar-Sensoren werden vor der weiteren Verarbeitung fusioniert. An dieser Stelle in der Verarbeitungskette sind auch Filter, wie beispielsweise ein Bodenfilter (*Ground Removal Filter*), implementierbar, die die anschließende Segmentierung erleichtern.

2.3.1 Segmentierung

Der erste Auswerteargorithmus der Verarbeitungskette ist die Segmentierung. Ziel der Segmentierung ist es, in der gesamten Punktwolke zusammenhängende homogene Regionen zu finden, deren Punkte ähnliche Charakteristiken aufweisen.^{29a} Jede dieser Regionen bildet ein eigenständiges Segment und beinhaltet eine Teilmenge der vollständigen Punktwolke. In diesem Schritt ist die Objektklasse des gefundenen Segments irrelevant. Bevor auf die Segmentierungsmethode beim Lidar-Sensormodell von FZD eingegangen wird, wird im Folgenden kurz ein Überblick über die verschiedenen Segmentierungskategorien gegeben. Die Segmentierung wird nach Nguyen und Le^{29b} in fünf Kategorien eingeteilt:

²⁹ Vgl. Nguyen, A.; Le, B.: 3D Point Cloud Segmentation (2013), a: S. 225; b: S. 226; c: S. 227f.; d: S. 228.

-
- kantenbasierte,
 - regionsbasierte,
 - attributsbasierte,
 - modellbasierte und
 - graphenbasierte Segmentierung (auch Segmentierung mittels maschinellen Lernens nach Grilli et al. genannt³⁰).

Kantenbasierte Segmentierung

Die kantenbasierte Segmentierung beruht auf der Annahme, dass bei einer starken Änderung der Intensität entlang einer Linie in der Punktwolke eine Kante vorliegt und diese ein Segment abgrenzt. Der Einsatz der Methode der kantenbasierten Segmentierung führt nach Nguyen und Le jedoch bei verrauschten Punktwolken oder Punktwolken mit ungleicher Punktedichte zu keinen guten Ergebnissen.^{29b}

Regionsbasierte Segmentierung

Dieses Problem umgeht die regionsbasierte Segmentierung, die räumlich nah beieinanderliegende Punkte mit ähnlichen Eigenschaften in einem Segment zusammenfasst und somit nach Unterschieden zwischen den Regionen in einer Punktwolke sucht. Eine Einteilung in *bottom-up*- (bestimmte Anzahl an Startpunkten, die zu Segmenten wachsen, wird vorgegeben) und *top-down*-Methoden (Anfang mit allen Punkten in einem Segment und stetige Trennung in mehrere Segmente, solange ein Grenzwert eines gewählten Gütekriteriums nicht unterschritten wird) kann vorgenommen werden, für eine ausführliche Erklärung hierfür sei auf die Literatur hingewiesen (Nguyen und Le^{29b}). Bei der regionsbasierten Segmentierung besteht das Problem, dass diese zu Über- und Untersegmentierung neigt.

Attributsbasierte Segmentierung

Einteilung aller Punkte innerhalb der gesamten Punktwolke in Regionen mit ähnlichen Attributen ist die Methode, die bei der attributsbasierten Segmentierung zur Anwendung kommt. Beispiele für diese Attribute sind Distanzen oder lokale Punktedichten. Limitiert ist dieses Verfahren der Segmentierung durch die Rechenzeit, die beim Segmentieren von Punktwolken mit hoher Punkteanzahl mit multidimensionalen Attributen benötigt wird.^{29c}

Modellbasierte Segmentierung

Die modellbasierte Segmentierung verfolgt den Ansatz, geometrisch einfache Formen wie beispielsweise Linien, Ebenen oder Zylinder innerhalb der Punktwolke zu identifizieren. Ein bekanntes Beispiel hierfür ist der RANSAC-Algorithmus (*Random Sample Consensus*) von Fischler und Bolles³¹, der als Grundlage für weitere Algorithmen der modellbasierten Segmentierung gilt.^{29d}

³⁰ Vgl. Grilli, E. et al.: Point Clouds Segmentation And Classification Algorithms (2017), S. 341.

³¹ Vgl. Fischler, M. A.; Bolles, R. C.: Random Sample Consensus (1981).

Graphenbasierte Segmentierung

Bei der graphenbasierten Segmentierung (Segmentierung mittels maschinellen Lernens) werden Punktwolken unter Verwendung verschiedener Algorithmen als Graphen modelliert. Der von Golovinskiy und Funkhouser entwickelte k-Nearest-Neighbors Algorithmus (k-NN-Algorithmus) erzeugt aus der Punktwolke einen 3D-Graphen und versucht so, den Vorder- vom Hintergrund der Punktwolke zu trennen.³² Die Ausführung des k-NN-Algorithmus erfordert jedoch vorherige Kenntnis darüber, wo sich die zu segmentierenden Objekte befinden. Neben dem k-NN-Algorithmus existieren noch weitere Segmentierungsalgorithmen, die auf maschinellem Lernen basieren, wie der FH-Algorithmus von Felzenszwalb und Huttenlocher.³³ Eine Übersicht hierzu ist in Nguyen und Le gelistet.^{29d}

Eine Untersuchung über die Segmentierungsmethoden in einer vorherigen Arbeit bei FZD von Stäcker hat ergeben, dass die regionsbasierte Segmentierung für den vorliegenden Fall einer Sensorsimulation eines 2,5-D Lidar-Sensors am besten geeignet ist.^{34a} Hierzu sei erwähnt, dass der ibeo Lux 2010® in die Kategorie der 2,5-D Lidar-Sensoren eingeordnet wird, da dieser über vier vertikale Kanäle verfügt. Bei einem einzeiligen Lidar-Sensor wird von einem 2-D Lidar-Sensor gesprochen, bei mehr als neun Kanälen handelt es sich um einen 3-D Lidar-Sensor.³⁵ Weiterhin wird in der Untersuchung über die Segmentierungsmethoden der Algorithmus von Kim et al. gewählt, da dieser speziell für die Anwendung bei 2,5-D Lidar-Sensoren entwickelt ist. Die Funktionsweise des Algorithmus wird im Folgenden kurz erläutert.

Bei der Erfassung des gesamten horizontalen Öffnungswinkels sind alle reflektierten Punkte nach aufsteigendem Azimutwinkel geordnet. Dies ist die Folge der kontinuierlichen Rotation des Spiegels (siehe Abschnitt 2.1.1). Ein neu erkannter Punkt wird demnach immer rechts der Punkte liegen (größerer Azimutwinkel), die in der gleichen Abtastperiode erkannt werden. Nun wird von diesem Punkt der euklidische Abstand zu den jeweils letzten erkannten Punkten pro Kanal berechnet, also insgesamt vier Abstände bei vier Kanälen. Der Punkt wird zu dem Segment des Punktes hinzugefügt, bei dem der berechnete Abstand am geringsten ist, sofern ein definierter Schwellwert unterschritten wird. Falls der Schwellwert bei allen vier Abständen nicht unterschritten wird, so erzeugt der Punkt ein neues Segment.³⁶

In der Masterthesis von Stäcker wird weiterhin die Erweiterung des Algorithmus von Kim et al. implementiert, die die Robustheit der Segmentierung erhöht, indem nicht nur die Abstände zu den jeweils letzten Punkten aller Kanäle berechnet werden, sondern die jeweils letzten beiden Punkte pro Kanal betrachtet werden. Außerdem wird der Schwellwert für die Abstandsberechnung in eine X- und Y-Komponente aufgeteilt, da bei einem Objekt, das sich mittig vor dem Fahrzeug befindet, die Punkte in X-Richtung deutlich dichter angeordnet sind als in Y-Richtung.^{34b}

³² Vgl. Golovinskiy, A.; Funkhouser, T.: Min-Cut Based Segmentation of Point Clouds (2009).

³³ Vgl. Felzenszwalb, P. F.; Huttenlocher, D. P.: Efficient Graph-Based Image Segmentation (2004).

³⁴ Vgl. Stäcker, L. S.: Masterthesis, Informationssystem zum Segmentieren und Tracking von Objekten (2017), a: S. 36; b: S. 40.

³⁵ Vgl. Kim, B. et al.: Pedestrian/Vehicle Detection Using a 2.5-D Multi-Layer Laser Scanner (2016), S. 400.

³⁶ Vgl. Kim, B. et al.: Robust object segmentation using a multi-layer laser scanner (2014), S. 20407.

Neben dem Segmentierungsalgorithmus von Kim et al. wird in einer weiteren Masterthesis bei FZD von Knerr ein zweiter Algorithmus zur Segmentierung implementiert. Der Radially-Bounded-Nearest-Neighbor-Algorithmus (RBNN-Algorithmus) von Klasing et al.³⁷ benötigt im Gegensatz zum zuvor beschriebenen Ansatz von Kim et al. keine Information über die Kanalzugehörigkeit der Punkte. Dies ist von Vorteil, da die Simulation der erstellten Szenarien mit VTD zum Zeitpunkt des Verfassens dieser Masterthesis noch nicht über die entsprechende Ausgabe verfügt, in welchem Kanal ein reflektierender Punkt erfasst wird. Des Weiteren benötigt der RBNN-Algorithmus weniger Parameter und Rechenzeit, da nicht bei jedem neuen Punkt der Abstand zu den jeweils letzten erkannten Punkten je Kanal berechnet wird.^{38a} Bei dem RBNN-Algorithmus wird für jeden Punkt in einem Suchradius nach benachbarten Punkten gesucht. Findet sich ein benachbarter Punkt, so wird der Punkt zu dessen Segment zugeordnet. Falls sich kein benachbarter Punkt im Suchradius befindet, so initiiert der Punkt ein neues Segment. Zusätzlich hat Knerr noch die Erweiterung von Choe et al. in den ursprünglichen RBNN-Algorithmus implementiert, bei der der Suchradius in Abhängigkeit vom Abstand des Punktes zum Sensor und der vertikalen Winkelauflösung des Sensors angepasst wird.³⁹ Für den Suchradius r_i ergibt sich in Abhängigkeit des Abstandes x_i mit einer vertikalen Winkelauflösung $\Phi_i = 0,8^\circ$

$$r_i = x_i \tan 0,8^\circ. \quad (2-8)$$

Der Suchradius wird mit einem Faktor r_{scale} skaliert für den Fall, dass nicht alle Reflexionspunkte eines Objektes in einer vertikalen Ebene liegen. Dieser Faktor wird im Ansatz von Knerr auf $r_{\text{scale}} = 3$ festgelegt, da dies dem maximalen Abstand zwischen den vier Kanälen bei maximaler Reichweite entspricht.^{38c} Eine Darstellung dieser Erweiterung befindet sich in Abbildung 2-8.

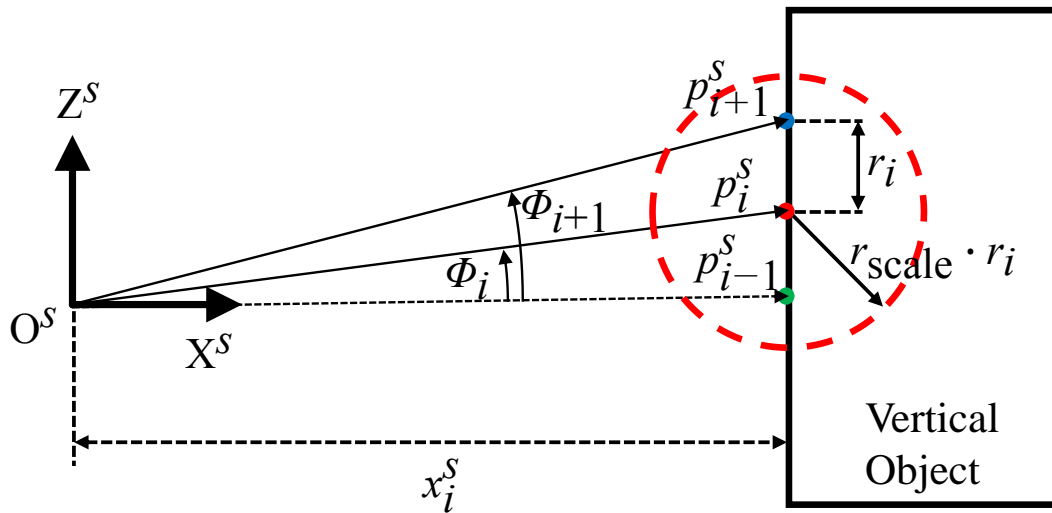


Abbildung 2-8: Adaption des Suchradius bei der Erweiterung des RBNN-Algorithmus⁴⁰

³⁷ Vgl. Klasing, K. et al.: A clustering method for efficient segmentation (2008).

³⁸ Vgl. Knerr, J.: Masterthesis, Entwicklung eines LiDAR-Tracking-Algorithmus für Sensormodelle (2018), a: S. 35; b: S. 37; c: S. 36.

³⁹ Vgl. Choe, Y. et al.: Fast Point Cloud Segmentation (2012), S. 40.

⁴⁰ Choe, Y. et al.: Fast Point Cloud Segmentation (2012), S. 40.

2.3.2 Tracking

Der zweite Auswertalgorithmus in Abbildung 2-6 ist das Tracking. Die Eingangsschnittstelle des Trackingalgorithmus bildet die segmentierte Punktwolke, die das Ergebnis und die Ausgabe des Segmentierungsalgorithmus ist. Granström teilt das Tracking, also die Zustandsschätzung von Objekten, in das *Point Object Tracking*, das *Extended Object Tracking*, das *Group Object Tracking* sowie das *Tracking with Multi-Path Propagation* (Mehrwegeausbreitung) ein.⁴¹ Beim *Point Object Tracking* existiert für jedes Objekt pro Zeitschritt maximal eine Reflexion. Die Objekte werden also als Punktmassen modelliert, die keine räumliche Ausdehnung haben. Bei dem *Extended Object Tracking* hingegen ruft jedes Objekt in einem Zeitschritt mehrere Reflexionen hervor, eine Berechnung der räumlichen Ausdehnung sowie der Orientierung des Objekts ist nach mehreren konsekutiven Messungen möglich. Beim *Group Object Tracking* wird eine Gruppe von mehreren Objekten ebenfalls von mehreren Auflösungszellen des Sensors detektiert, der Unterschied zum *Extended Object Tracking* besteht jedoch darin, dass diese Gruppe aus individuellen Subobjekten besteht, die sich gemeinsam bewegen. *Tracking with Multi-Path Propagation* bedeutet das Verfolgen von Objekten, bei dem jedes Objekt mehrere Reflexionen erzeugt, die sich unter anderem aus der Mehrwegeausbreitung ergeben. Der Unterschied zum *Extended Object Tracking* liegt in der Verteilung der Punkte, die beim *Tracking with Multi-Path Propagation* aufgrund möglicher Scheinobjekte nicht räumlich strukturiert ist.⁴¹

Im Lidar-Sensormodell von FZD ist ein *Extended Object Tracker* implementiert. Dieser besitzt als Eingangsschnittstelle direkt die Punktwolke und integriert die Segmentierung. Die Funktion dieses Trackingalgorithmus wird anhand der Abbildung 2-9 erläutert. Der erste Schritt ist die Sensordatenverarbeitung, in der die Daten des Sensors vorverarbeitet werden. Dieser Schritt umfasst neben dem Filtern von Bodenreflexionen auch die Segmentierung, die im vorherigen Abschnitt ausführlich beschrieben ist. Beim Schritt der Datenassoziation werden die neuen Messwerte den bereits vorhandenen Objekthypothesen zugeordnet. Hierfür existieren mehrere Verfahren, für das Lidar-Sensormodell von FZD wurde das *Nearest-Neighbor*-Verfahren gewählt. Jedem bestehenden Objekt wird die Messung zugeordnet, deren euklidische Distanz zum bestehenden Objekt am geringsten ist. Wenn für eine Messung noch kein prädiiziertes Objekt besteht, so initiiert diese Messung ein neues Objekt. Eine weitere Abfrage verhindert das mehrfache Hinzufügen einer einzelnen Messung zu verschiedenen bestehenden Objekten.^{38b}

⁴¹ Vgl. Granström, K. et al.: *Extended Object Tracking* (2016), S. 2f.

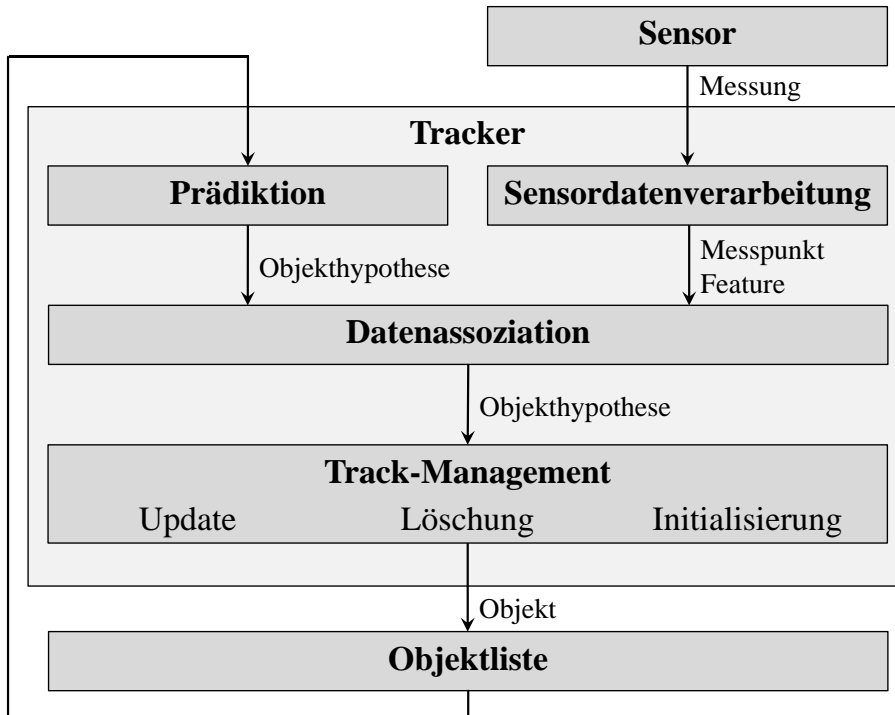


Abbildung 2-9: Struktur eines Trackingalgorithmus⁴²

Mittels eines Kalman-Filters wird im Updateschritt des Track-Managements der Zustand jedes Objekts, welches von der Datenassoziation an das Track-Management übergeben wird, aktualisiert. Jedes Objekt hat zum Zeitpunkt k folgende Zustände:

$$\mathbf{O}_k = [x_k, y_k, z_k, v_{rel,k}, v_{abs,k}, \Psi_k, l_k, w_k, h_k]^T. \quad (2-9)$$

Dabei repräsentieren x_k , y_k , und z_k die Position des Schwerpunktes und des Mittelpunkts, $v_{rel,k}$ und $v_{abs,k}$ die Relativ- und Absolutgeschwindigkeit, Ψ_k den Gierwinkel und l_k , w_k , h_k die Abmessungen des Objekts. Zusätzlich wird jedem Objekt eine eindeutige ID zugeordnet. Erst wenn ein Objekt in mehr als fünf konsekutiven Messungen erfasst wird (Parameter *Lifetime* > 5), so wird es initialisiert. Falls es in mehr als zehn konsekutiven Messungen nicht mehr erkannt wird (Parameter *Lost* > 10), wird das Objekt aus dem Track-Management gelöscht.

2.3.3 Klassifizierung

Die Klassifizierung ist der letzte Schritt der Auswertelgorithmen (siehe Abbildung 2-6). Grilli et al. beschreiben die zuvor ausgeführte Segmentierung als den Prozess, bei dem aus der Punktwolke Regionen gebildet werden, die jeweils eine oder mehrere gleiche Eigenschaften besitzen. Die Klassifizierung hingegen wird als das Zuweisen von spezifischen Klassen zu den einzelnen Punkten, die

⁴² Mit Änderungen entnommen aus Knerr, J.: Masterthesis, Entwicklung eines LiDAR-Tracking-Algorithmus für Sensormodelle (2018), S. 18.

vorher segmentiert werden, definiert.⁴³ Auch in weiteren Veröffentlichungen folgt die Klassifizierung direkt dem Segmentierungsschritt.^{44,45,46}

In einer vorherigen Masterthesis am Fachgebiet Fahrzeugtechnik der TU Darmstadt von Ngo^{47a} wurden zwei Klassifizierungsalgorithmen für das Lidar-Sensormodell entwickelt und implementiert. Beide Algorithmen verfolgen dabei einen unterschiedlichen Ansatz. Mit der End-to-End-Objekterkennung werden direkt die Rohdaten ohne eine vorherige Segmentierung klassifiziert. Die Grundlage dieser Objekterkennung bilden *Convolutional Neural Networks*, also Deep-Learning-Algorithmen. Ngo schlägt hierfür die Verwendung des generischen 3D-Objekterkennungsnetzes *VoxelNet*^{48a} vor. Dieses ist in der Lage, die Klassen „Fahrzeug“, „Fußgänger“ und „Zweiradfahrer“ aus der Punktwolke zu extrahieren (siehe Abbildung 2-10). Die Evaluation der Implementierung von VoxelNet hat jedoch ergeben, dass die Ergebnisse von Zhou und Tuzel nicht reproduziert werden können. Dies hat mehrere Ursachen, unter anderem auch, dass Zhou und Tuzel die Eingangsdaten des VoxelNet mit einem Lidar-Sensor mit 64 Kanälen erzeugen.^{47b} Da der bei FZD simulierte Lidar-Sensor nur über vier Kanäle verfügt, erzeugt dieser auch deutlich weniger Informationen über die dreidimensionale Ausdehnung der Objekte.⁴⁹

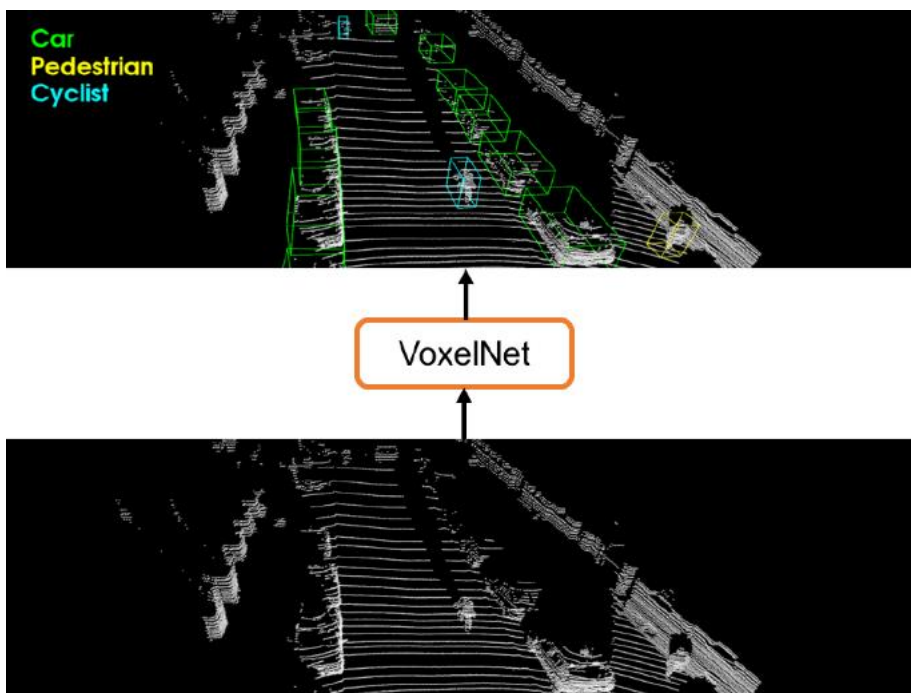


Abbildung 2-10: Objekterkennung mit VoxelNet^{48b}

⁴³ Vgl. Grilli, E. et al.: Point Clouds Segmentation And Classification Algorithms (2017), S. 339.

⁴⁴ Vgl. Yokoyama, H. et al.: Detection and Classification of Pole-like Objects (2013), S. 2.

⁴⁵ Vgl. Douillard, B. et al.: Segmentation and Classification of 3D Point Clouds (2014), S. 585.

⁴⁶ Vgl. Aijazi, A. et al.: Segmentation Based Classification (2013), S. 1626.

⁴⁷ Vgl. Ngo, A.: Masterthesis, Simulationsbasiertes Testen und Entwickeln von Objekterkennungsalgorithmen (2018), a: -; b: S. 47f.

⁴⁸ Vgl. Zhou, Y.; Tuzel, O.: VoxelNet (2017), a: -; b: S. 1.

⁴⁹ Vgl. Holder, M. et al.: Requirements for a Lidar Sensor Model (2018), S. 3.

Der zweite Ansatz zur Implementierung eines Klassifikationsalgorithmus ist die mehrstufige Objekterkennung, die an die in Abschnitt 2.3.1 beschriebene Segmentierung anschließt. Der Klassifizierungsalgorithmus basiert auf einem *Machine-Learning*-Ansatz. Ausgehend von der segmentierten Punktwolke ist der erste Schritt der Objekterkennung die Merkmalsextraktion. In diesem Schritt werden aus den einzelnen Teilpunktwolken definierte Merkmale wie z. B. die Abmessungen des Segments, die Standardabweichung der Punkte in den Raumrichtungen oder die Entfernung des Segments zum Sensor extrahiert. Eine Übersicht über die Merkmale, die für jedes Segment bestimmt werden, befindet sich in Tabelle 2-2. Die Klassen unterscheiden sich hauptsächlich durch die unterschiedlichen Merkmale. Eine Untersuchung von Stäcker hat ergeben, dass einige der Merkmale entfernt werden können, ohne dass die Genauigkeit der Klassifizierung abnimmt. Die reduzierten Merkmale kommen ohne die Merkmale Distanz, Höhe, Standardabweichung der Z-Koordinate und Punkte pro Distanz aus (die nicht benötigten Merkmale sind in Tabelle 2-2 hell hinterlegt).⁵⁰

Die Einteilung der Segmente in die Klassen erfolgt mittels eines Entscheidungsbaumes. Ein *Bagged Decision Tree* mit 100 Entscheidungsbäumen und maximal 20 Abzweigungen je Entscheidungsbaum wird implementiert. Der Ausgang der Klassifizierung ist eine Vorhersage über die Klassenzugehörigkeit der Segmente in der Objektliste.

Tabelle 2-2: Merkmale nach Stäcker⁵⁰

Merkm	Beschreibung
length	Naive Länge, X-Richtung
width	Naive Breite, Y-Richtung
height	Naive Höhe, Z-Richtung
meanX	Mittlere Entfernung in X-Richtung
meanY	Mittlere Entfernung in Y-Richtung
meanZ	Mittlere Entfernung in Z-Richtung
stdX	Standardabweichung der X-Werte
stdY	Standardabweichung der Y-Werte
stdZ	Standardabweichung der Z-Werte
distance	Distanz zum Mittelpunkt
np	Anzahl der Punkte
np/distance	Anzahl der Punkte pro Distanz
np·distance	Anzahl der Punkte multipliziert mit der Distanz

⁵⁰ Vgl. Stäcker, L. S.: Masterthesis, Informationssystem zum Segmentieren und Tracking von Objekten (2017), S. 51.

3 Analyse möglicher Fehler in der Lidar-Sensorsimulation mittels FMEA

In diesem Kapitel wird die in Abschnitt 2.3 vorgestellte Verarbeitungskette der Lidar-Sensorsimulation analysiert und bewertet. Das Analysieren der möglichen Fehler und die Bewertung erfolgt mittels einer (Software-)FMEA (Fehlzustandsart- und -auswirkungsanalyse, engl. *Failure Mode and Effect Analysis*) und orientiert sich an dem Entwurf der gültigen Norm DIN EN 60812 aus dem Jahr 2015. Das Ziel einer FMEA ist es, alle möglichen Ausfallarten sowie deren Ursachen und Auswirkungen eines Prozesses (oder einer Software) zu identifizieren. Eine FMEA dient dazu, die ermittelten Ursachen zu analysieren und die negativen Auswirkungen auf den Prozess zu minimieren oder eliminieren. Außerdem ergibt sich die Möglichkeit, die Kritikalität der Ausfallarten zu bewerten, um die Prioritäten bei der Entwicklung des Prozesses (oder der Software) festzulegen.⁵¹

Das Ziel dieses Kapitels ist es, zu überprüfen, ob eine FMEA für die Bewertung einer (Lidar-)Sensorsimulation geeignet ist. Die FMEA ist nach DIN EN 60812 in drei Phasen unterteilt, es wird zwischen der Planungs-, Durchführungs- und Berichtsphase unterschieden. Die Berichtsphase entspricht diesem Kapitel. Im Folgenden werden die einzelnen Phasen erläutert und parallel dazu die Rolle der Phasen für die Bewertung der Lidar-Sensorsimulation erklärt.

Für den ersten Schritt der Planungsphase der FMEA werden der Bereich bzw. die Module definiert, auf die die FMEA angewendet wird. Außerdem werden die Grenzen und Umgebungsbedingungen des analysierten Prozesses oder der Software festgelegt. Im Falle der FMEA für die Lidar-Sensorsimulation beschränkt sich die FMEA auf die Software, daher handelt es sich um eine Software-FMEA. Untersucht wird die gesamte Software, die einen Teil der Lidar-Sensorsimulation darstellt. Das erste Modul ist die Simulationsumgebung VTD, in der sich das Ego-Fahrzeug mit den Lidar-Sensoren bewegt und in dem die Punktwolken generiert werden. Abgegrenzt wird dieses Modul von der Szenarienerstellung im Szenario-Editor sowie der Modellierung der Umwelt im Road Designer als Eingang und der Ausgabe der Punktwolke als Ausgang. Die weiteren Module bestehen aus den Auswertelgorithmen. Das zweite Modul stellt die Segmentierung (S) dar. Zum dritten Modul ist dieses Modul abgegrenzt durch die segmentierte Punktwolke. Die Schnittstelle dieses Moduls zum ersten Modul wird durch die Punktwolke gebildet. Das dritte Modul ist das Tracking (T), welches die Segmente des zweiten Moduls als Eingang besitzt und eine Liste mit getrackten Segmenten ausgibt. Die Klassifizierung (K) bildet das vierte und letzte Modul. Durch die Ausgabe der getrackten und klassifizierten Objekte in der sogenannten Objektliste ist das letzte Modul abgegrenzt.

Im zweiten Schritt der Planungsphase einer FMEA werden Entscheidungskriterien für die Behandlung von Ausfallarten definiert. Für die Lidar-Sensorsimulation bedeutet dies konkret, dass zunächst alle möglichen Abweichungen zwischen Realität und Simulation von Bedeutung sind. Die Schwere der Ausfallwirkung, die Wahrscheinlichkeit, dass die Ausfallart auftritt und die Wahrscheinlichkeit, dass die Auswirkung eintritt, werden in diesem Schritt noch nicht berücksichtigt. Ein weiterer Schritt in der Planungsphase ist die Definition von Ressourcen für die Analyse. Es werden alle Informationen, die über die Auswertelgorithmen vorhanden sind, verwendet. Dies umfasst hauptsächlich die

⁵¹ Vgl. DIN Deutsches Institut für Normung e. V.: DIN EN 60812 (2015), S. 10.

studentischen Arbeiten, die die einzelnen Algorithmen und die Implementierung beschreiben. Für die Ressourcen zum Simulationstool VTD wird auf das VTD-Wiki sowie die Konfigurationsdateien zurückgegriffen.

Bei der Durchführungsphase der FMEA wird zwischen der *Top-down*- und der *Bottom-up*-Analyse unterschieden. Die *Top-down*-Analyse betrachtet jede Funktion und die Ausfallarten dieser Funktion auf höchster Ebene nacheinander bis zur untersten Ebene. Bei der *Bottom-up*-Analyse werden die Auswirkungen der Ausfallarten in den unteren Ebenen und deren Einfluss auf die höheren Ebenen bestimmt. Für die FMEA der Lidar-Sensorsimulation wird der *Top-down*-Ansatz gewählt. Mögliche Fehler, die in den Ausgängen der Module auftreten können, werden aufgelistet und deren Ursprung sowie deren Auswirkung auf nachgelagerte Module analysiert. In Tabelle 3-1 befindet sich die FMEA zur Lidar-Sensorsimulation, die im Folgenden erläutert wird.

In der zweiten Spalte der Tabelle sind die vier zuvor definierten Module aufgelistet (VTD, S, T, K). Die dritte Spalte beinhaltet die möglichen Fehlverhalten des Ausgangs des jeweiligen Moduls, diese sind mit einer fortlaufenden Nummer in Spalte eins versehen. Eine Auflistung der verantwortlichen Software (SW)-Komponente oder des verantwortlichen vorherigen Moduls für das Fehlverhalten des betrachteten Moduls befindet sich in der vierten Spalte. Hierin ist die SW-Komponente ein Teil des betrachteten Moduls, während das vorherige Modul alle möglichen vorherigen Module beinhalten kann. Bei Betrachtung des ersten Moduls existiert kein vorheriges Modul, somit stammen alle Fehlverhalten des Moduls von einzelnen SW-Komponenten dieses Moduls. In der fünften Spalte ist das spezielle Fehlverhalten der SW-Komponenten aufgelistet. Falls das Fehlverhalten auf ein vorheriges Modul zurückzuführen ist, stehen an dieser Stelle das Fehlverhalten des vorherigen Moduls sowie ein Verweis zu diesem. Die sechste Spalte beinhaltet die Auswirkungen des Fehlverhaltens auf die nachfolgenden Module sowie ein Verweis auf das nachfolgende Modul, falls vorhanden. Da bei dem Klassifizierungsmodul kein nachfolgendes Modul existiert, sind die entsprechenden Zellen in dieser Spalte leer. In der letzten Spalte sind mögliche Lösungsansätze aufgelistet zur Behebung des Fehlverhaltens der Komponente, das zum Fehlverhalten des Moduls führt.

Die vorgestellte FMEA erhebt keinen Anspruch auf Vollständigkeit und veranschaulicht nur beispielhaft, ob eine Sensorsimulation mittels FMEA bewertet werden kann. Im Folgenden wird auf die Herausforderung bei der Erstellung der FMEA hingewiesen. Zunächst wird für die Erstellung einer FMEA ein Expertenteam benötigt, dessen Mitglieder jeweils mit den einzelnen Modulen vertraut sind und einschätzen können, welche Auswirkung ein Fehler in der vorherigen Verarbeitungskette auf das vorliegende Modul besitzt. Daher existiert in dieser FMEA keine Spalte mit der Angabe der Schwere eines Fehlverhaltens. Außerdem sind Häufigkeiten des Auftretens eines Fehlverhaltens nicht explizit zu ermitteln, da viele Fehlverhalten eine Parametrisierung erfordern, die wiederum einem Optimierungsproblem unterliegt. Abschließend sei gesagt, dass eine FMEA zur Bewertung einer Sensorsimulation grundsätzlich möglich ist, jedoch mit den genannten Einschränkungen. Die Zusammenhänge der Fehler der unterschiedlichen Verarbeitungsebenen werden dennoch verdeutlicht.

Tabelle 3-1: FMEA zur Lidar-Sensorsimulation

Nr.	Modul	Fehlverhalten des Ausgangs des Moduls	Für Fehlverhalten verantwortliche SW-Komponente/n oder vorheriges Modul	Fehlverhalten der Komponente oder des vorherigen Moduls	Auswirkungen auf nachfolgende Module	Lösungsvorschlag
10	VTD	Simulierte Punktwolke hat einen Versatz oder ist rotiert	Lidar-Sensor	Falsche Positionierung oder Orientierung der Sensoren am Ego-Fahrzeug	Objektliste: Position und Orientierung der Objekte falsch	Parametrierung der Sensoren anpassen
20	VTD	Anzahl der simulierten Punkte ist zu hoch/zu niedrig	Lidar-Sensor	Falsche Winkelauflösung (horizontal, vertikal) oder falscher maximaler Öffnungswinkel	Segmentierung: Über-/Untersegmentierung, siehe Nr. 60 Objektliste: Objekte fehlen oder haben falsche Abmessungen in der Objektliste	Winkelauflösung und maximaler Öffnungswinkel in Parameterdatei anpassen
				Falsche Längenauflösung (radial) oder falsche maximale Reichweite		Längenauflösung und maximale Reichweite in Parameterdatei anpassen
				Raytracing/Raylaunching berücksichtigt keine Mehrfachreflexionen	Segmentierung: Über-/Untersegmentierung, siehe Nr. 60 Objektliste: Keine Scheinziele vorhanden	Mehrfachreflexion in Modell implementieren
			Umgebungsmodell	Rauigkeiten nicht berücksichtigt	Segmentierung: Über-/Untersegmentierung, siehe Nr. 60	Rauigkeiten berücksichtigen
				Falsche Rauigkeiten den Objekten zugewiesen		Objekten die richtigen Rauigkeiten zuweisen
			Umgebungsmodell	Infrastruktur (z. B. Schutzplanken, Beschilderung, Lichtsignalanlagen) nicht modelliert	Rückwirkung auf Punktwolke: Falls Mehrfachreflexion implementiert ist, führt nicht vorhandene Infrastruktur zu weniger Reflexionen und somit zu den gleichen Auswirkungen wie „Raytracing/Ray-launching berücksichtigt keine Mehrfachreflexionen“.	

Nr.	Modul	Fehlverhalten des Ausgangs des Moduls	Für Fehlverhalten verantwortliche SW-Komponente/n oder vorheriges Modul	Fehlverhalten der Komponente oder des vorherigen Moduls	Auswirkungen auf nachfolgende Module	Lösungsvorschlag
30	VTD	Geometrie der simulierten Punktwolke stimmt nicht mit Geometrie der realen Fahrzeuge überein	Umgebung	Fahrzeugmodelle entsprechen nicht der Realität in Bezug auf die Geometrie	Segmentierung: Über-/Untersegmentierung, da z. B. Reflexionspunkte im Inneren des Fahrzeugs auftreten, siehe Nr. 60 Tracking: Objektgrößen werden falsch geschätzt	Fahrzeugmodelle mit korrekter Geometrie implementieren
				Fahrzeugmodelle entsprechen nicht der Realität in Bezug auf Materialien und besitzen dadurch falsche Reflektivitäten, z. B. die Fenster der Fahrzeuge sind nicht korrekt modelliert		Korrekte Reflektivitätswerte den Materialien der Fahrzeugmodelle zuweisen
				Fahrzeugmodelle besitzen geringeren Detaillierungsgrad, z. B. keine Kennzeichen mit höherer Reflektivität modelliert		Wichtige Bauteile in Bezug auf Reflektivität in Fahrzeugmodellen berücksichtigen
40	VTD	Vertikaler Versatz der Objekte in der simulierten Punktwolke	Umgebung	Höhenprofil des Umgebungsmodells entspricht nicht der Realität	Objektliste: Position der Objekte in Objektliste falsch	Höhenprofil der Umgebung im Road Designer einfügen
50	VTD	Objekte in der simulierten Punktwolke sind an falschen Positionen und/oder rotiert	Szenarienerstellung	Falsche Positionen, Orientierung und/oder Geschwindigkeiten der Fahrzeuge im Szenario-Editor	Objektliste: Position, Orientierung und/oder Geschwindigkeit der Objekte in Objektliste falsch	Bei Szenarienerstellung darauf achten, dass die Position/Orientierung/Geschwindigkeit der Fahrzeuge korrekt ist. Falls ADMA-Daten eingelesen werden, diese vorher validieren.

Nr.	Modul	Fehlverhalten des Ausgangs des Moduls	Für Fehlverhalten verantwortliche SW-Komponente/n oder vorheriges Modul	Fehlverhalten der Komponente oder des vorherigen Moduls	Auswirkungen auf nachfolgende Module	Lösungsvorschlag
60	S	Über-/Untersegmentierung	Bodenfilter	Schwellwert des Bodenfilters falsch parametrisiert	Tracking: Trackkontinuität weist Lücken auf	Geeigneten Schwellwert wählen
			Segmentierungsalgorithmus von Knerr	Skalierungsfaktor r_{scale} zu groß/zu klein gewählt	Objektliste: Zu viele/zu wenig Objekte in der Objektliste	Geeigneten Skalierungsfaktor wählen
			Segmentierungsalgorithmus von Stäcker	Anzahl der berücksichtigten Kanäle für die Suche nach benachbarten Punkten zu groß/zu klein gewählt		Geeignete Anzahl der berücksichtigten Kanäle für die Suche nach benachbarten Punkten wählen
			Segmentierungsalgorithmus von Stäcker	Schwellwerte in X- und Y-Richtung für die Suche nach benachbarten Punkten zu groß/zu klein gewählt		Geeignete Schwellwerte in X- und Y-Richtung für die Suche nach benachbarten Punkten wählen
			Segmentierungsalgorithmus von Stäcker	Schwellwert für die Vereinigung von benachbarten Segmenten zu groß/zu klein gewählt		Geeigneten Schwellwert für die Vereinigung von benachbarten Segmenten wählen
				Siehe Modul VTD Nr. 20: Anzahl der simulierten Punkte ist zu hoch/zu niedrig		Siehe Modul VTD Nr. 20
				Siehe Modul VTD Nr. 30: Geometrie der simulierten Punktwolke stimmt nicht mit Geometrie der realen Fahrzeuge überein		Siehe Modul VTD Nr. 30

Nr.	Modul	Fehlverhalten des Ausgangs des Moduls	Für Fehlverhalten verantwortliche SW-Komponente/n oder vorheriges Modul	Fehlverhalten der Komponente oder des vorherigen Moduls	Auswirkungen auf nachfolgende Module	Lösungsvorschlag
70	T	Getrackte Segmente werden nicht oder zu spät in die getrackte Segmentliste aufgenommen	Track-Management	Parameter <i>Lifetime</i> zu groß/zu klein gewählt	Objektliste: Zu wenig Objekte in der Objektliste	Geeigneten Parameter <i>Lifetime</i> wählen
80	T	Getrackte Segmente werden zu früh/zu spät/nicht von der getrackten Segmentliste gelöscht	Track-Management	Parameter <i>Lost</i> zu groß/zu klein gewählt	Objektliste: Zu wenig/zu viele Objekte in der Objektliste	Geeigneten Parameter <i>Lost</i> wählen
90	T	Abmessung der Fahrzeuge stimmen nicht mit den realen Abmessungen überein	Modul: VTD	Siehe Modul VTD Nr. 30: Geometrie der Punktwolke stimmt nicht mit Geometrie der realen Fahrzeuge überein	Objektliste: Fahrzeugabmessungen stimmen nicht	Siehe Modul VTD Nr. 30
100	K	Objekte in Objektliste falsch klassifiziert	Allgemein	Mögliche Klassen unterscheiden sich von den möglichen Klassen des realen Sensors	-	Alle Klassen berücksichtigen, die im realen Sensor berücksichtigt werden
			Bagged Decision Tree	Anzahl der Entscheidungsbäume und Anzahl der Abzweigungen je Entscheidungsbaum zu groß/zu klein gewählt	-	Anzahl der Entscheidungsbäume und Anzahl der Abzweigungen je Entscheidungsbaum anpassen
			Training	Anzahl der ausgesuchten Merkmale zu gering oder Merkmale führen in speziellen Fällen zu Falschklassifikation	-	Merkmale überprüfen, Merkmale hinzufügen
			Training	Trainierte Daten decken nicht alle Anwendungsfälle ab	-	Neue Trainingsdaten mit ausgewogenerem Objektdatensatz erzeugen

4 Übersicht Validierungsmethoden und angewandte Methodik

In diesem Kapitel wird die Methodik beschrieben, mit der die Bewertung der Lidar-Sensorsimulation durchgeführt wird. Die umfassende Literaturrecherche zeigt, dass im Bereich der Bewertung von Sensorsimulationen nur wenige Publikationen veröffentlicht sind. Dies liegt unter anderem daran, dass die Entwicklung von Sensorsimulationen für Umfelderkennungssensorik allgemein – und speziell die Validierung und Bewertung dieser – noch am Anfang steht. Die Methodik, auf der diese Masterthesis aufbaut, leitet sich aus den Veröffentlichungen von Schärmann et al.⁵², Roth et al.⁵³ und Oberkampff und Trucano^{54a} ab. Die vorgeschlagenen Ansätze dieser Veröffentlichungen werden dabei so angepasst, dass die Anwendung der Methoden für die in dieser Masterthesis relevanten Lidar-Sensorsimulation möglich ist.

Oberkampff und Trucano grenzen in ihrer Publikation die Begriffe Verifikation und Validierung voneinander ab. Verifikation ist demnach der „Prozess zur Bestimmung, ob eine Modellimplementierung die konzeptionelle Beschreibung des Modells des Entwicklers und eine Lösung des Modells korrekt widerspiegelt“⁵⁵. Die Validierung hingegen ist der „Prozess zur Bestimmung des Grades, zu dem ein Modell eine genaue Darstellung der realen Welt aus der Perspektive der beabsichtigten Verwendung des Modells ist“⁵⁵. In dieser Masterthesis hingegen werden Metriken entwickelt, die eine Bewertung des Sensormodells ermöglichen. Außerdem liegen keine Anforderungen vor, die die Voraussetzung einer Validierung sind. Aus diesem Grund und mit der Definition der Begriffe Verifizierung und Validierung wird die Aufgabe dieser Masterthesis nicht als Validierung eingestuft. Trotzdem wird teilweise auf die Validierungsmethodik von Oberkampff und Trucano sowie von Roth et al. zurückgegriffen, da eine Abwandlung dieser eine Bewertung des Lidar-Sensormodells ermöglicht.

Oberkampff und Trucano teilen den allgemeinen Validierungsprozess in drei Schritte ein, die im Folgenden anhand der Abbildung 4-1 erklärt werden. Der erste Schritt ist der Vergleich des Simulationsmodells mit experimentellen Daten mittels einer Validierungsmetrik. Im zweiten Schritt wird das Modell so inter- oder extrapoliert, dass die Bedingungen denen entsprechen, die im vorgesehenen Verwendungszweck vorliegen. Der dritte und letzte Schritt ist die Entscheidung, ob das Modell den geforderten Genauigkeitsanforderungen genügt.^{54b}

⁵² Vgl. Schaermann, A. et al.: Validation of Vehicle Environment Sensor Models (2017).

⁵³ Vgl. Roth, E. et al.: Analysis and Validation of Perception Sensor Models (2011).

⁵⁴ Vgl. Oberkampff, W. L.; Trucano, T. G.: Verification and validation benchmarks (2008), a: -; b: S. 724.

⁵⁵ Oberkampff, W. L.; Trucano, T. G.: Verification and validation benchmarks (2008), S. 719.

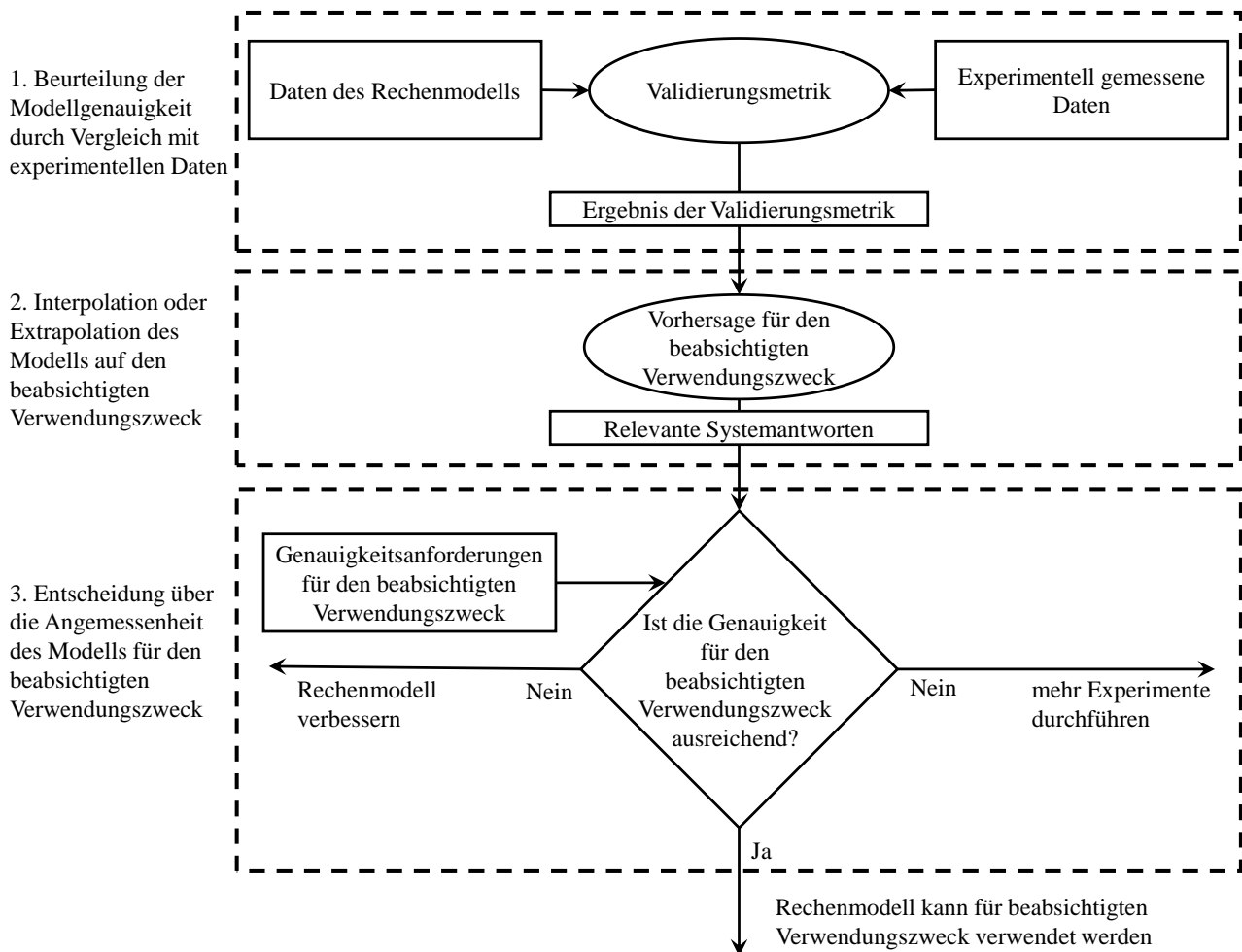


Abbildung 4-1: Allgemeine Validierungsmethode nach Oberkamp und Trucano⁵⁶

Roth et al. schlagen eine Methode für das Analysieren und Validieren von Sensormodellen vor (siehe Abbildung 4-2). Der erste Schritt der Methode ist das Durchführen von realen Testfahrten mit definierten Testmanövern und gleichzeitige Aufzeichnung der Positionsdaten und Daten der Umfelderkennungssensorik. In Abhängigkeit von den realen Positionsdaten werden im zweiten Schritt automatisch VTD-Szenarien generiert. Die Parametrierung des Sensormodells in VTD erfolgt gemäß den Parametern der realen Sensoren im dritten Schritt. Daran anschließend werden im vierten Schritt die virtuellen Testfahrten in VTD durchgeführt und in einem fünften Schritt die synthetischen Sensordaten aufgezeichnet. Die Validierung und Evaluation des Vergleichs zwischen den realen und den simulierten Daten bilden den letzten Schritt.

Roth et al. wenden die entwickelte Validierungsmethode an zwei beispielhaften Szenarien an. Bei den Sensormodellen, die in VTD implementiert werden, handelt es sich um ideale Sensormodelle. Der letzte Schritt der vorgeschlagenen Validierungsmethode erfolgt durch einen qualitativen Ver-

⁵⁶ Mit Änderungen entnommen aus Oberkamp, W. L.; Trucano, T. G.: Verification and validation benchmarks (2008), S. 725.

gleich von Diagrammen, in denen die Position und relative Geschwindigkeit von Objekten im Sichtfeld des Sensors aufgetragen sind. Ein quantitativer Vergleich unter Anwendung von Validierungsmetriken wird nicht durchgeführt.

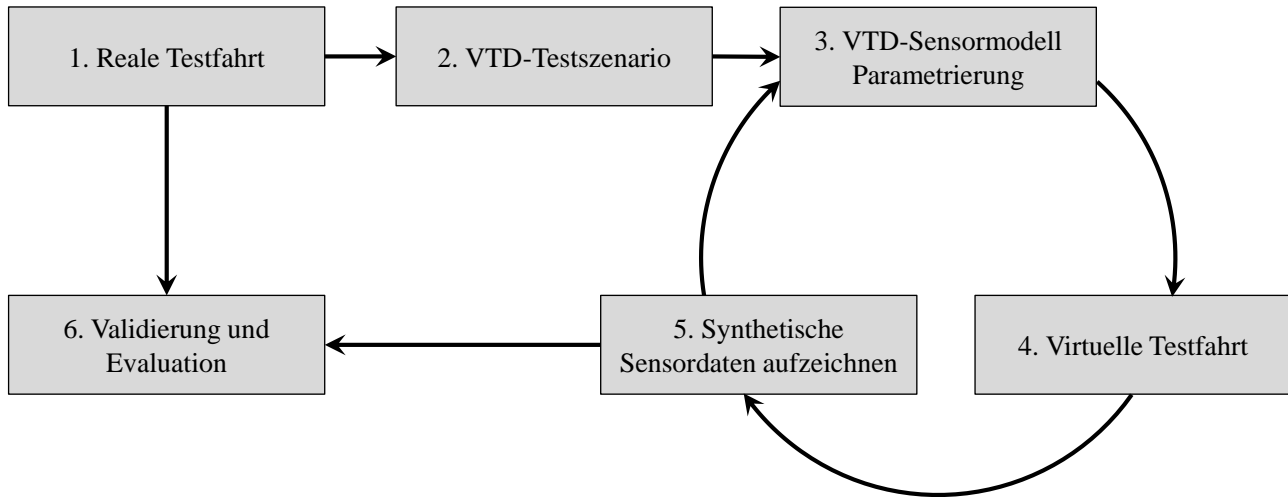


Abbildung 4-2: Validierung von Sensormodellen nach Roth et al.⁵⁷

Schärmann et al. stellen heraus, dass die Ansätze von Oberkampff und Trucano und Roth et al. zwar beide die Fragen zur Modellvalidierung aufgreifen, deren Fokus jedoch unterschiedlich ist.⁵⁸ Des Weiteren schlagen Schärmann et al. vor, die beiden Ansätze zu kombinieren. In der Kombination der Ansätze dient die Validierungsmethode von Roth et al. als Prozesskette, die beschreibt, wie reale und synthetische Daten beschafft werden. Nach Oberkampff und Barone⁵⁹ ist ein qualitativer Vergleich von realen und synthetischen Daten anhand von Diagrammen, wie Roth et al. ihn durchführen, jedoch für eine Validierung nicht ausreichend. Aus diesem Grund werden die mittels der Methode von Roth et al. erzeugten Daten in einem anschließenden Schritt mit quantitativen Metriken bewertet. Der Ausgang der Methode von Roth et al. dient also dem Eingang der Validierungsmethode von Oberkampff und Trucano.

Für die Validierung von Sensormodellen schlagen Schärmann et al. eine neue Methode vor, die auf den Untersuchungen von Oberkampff und Trucano, Oberkampff und Barone und Roth et al. basiert. Die Methode ist in sieben Schritte aufgeteilt, die im Folgenden anhand der Abbildung 4-3 erläutert werden. Der erste und zweite Schritt beziehen sich auf die Datenaufzeichnung im realen Versuchsfahrzeug. Mit den Sensoren werden Daten auf zwei verschiedenen Ebenen erfasst, entweder als Objektliste (OL) oder unverarbeitete Daten (Raw). Der Index R steht hier für reale Daten. Das Umfeldmodell stellt die Fusionierung mehrerer gleicher oder unterschiedlicher Sensortypen dar. Es wird zwischen der *High Level Fusion* (HLF) und *Low Level Fusion* (LLF) unterschieden. LLF bezeichnet

⁵⁷ Mit Änderungen entnommen aus Roth, E. et al.: Analysis and Validation of Perception Sensor Models (2011), S. 3.

⁵⁸ Vgl. Schaermann, A. et al.: Validation of Vehicle Environment Sensor Models (2017), S. 406.

⁵⁹ Vgl. Oberkampff, W. L.; Barone, M. F.: Measures of agreement between computation and experiment (2006), S. 6.

die Fusionierung von unverarbeiteten Daten, die meist aus mehreren Sensoren des gleichen Sensortyps stammen, während bei der HLF die OL von unterschiedlichen Sensortypen fusioniert werden.⁶⁰

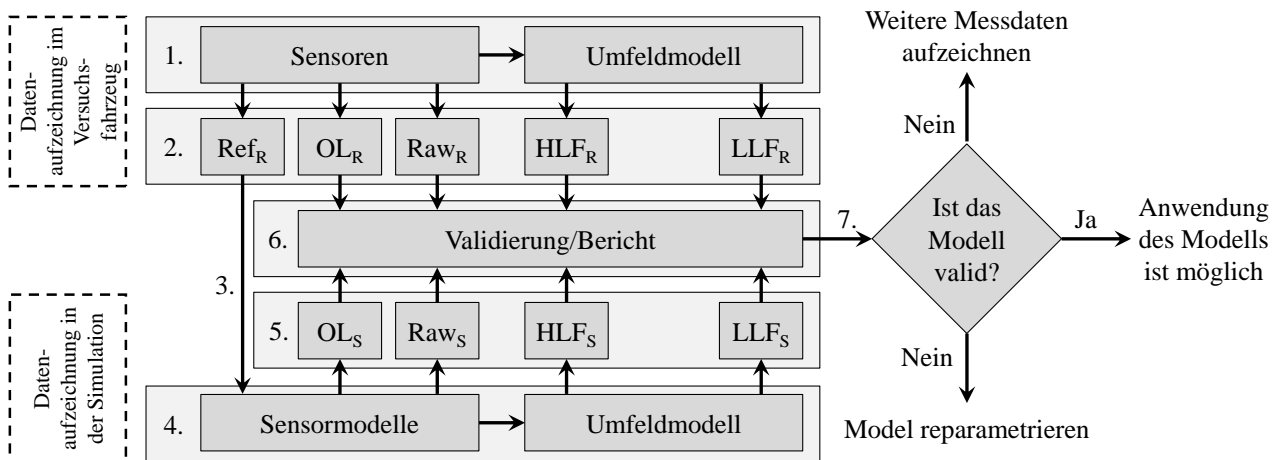


Abbildung 4-3: Validierung von Sensormodellen nach Schärman et al.⁶¹

Der dritte Schritt ist die Verknüpfung zwischen der realen und synthetischen Datenaufzeichnung. Die Referenzdaten der realen Testfahrt (Ref_R) werden an die Simulation übergeben, damit die Testszenarien nachgestellt werden können. Schritt vier und fünf sind analog zum ersten und zweiten Schritt, mit dem Unterschied, dass die Szenarien in der Simulationsumgebung virtuell abgefahren werden. Schärman et al. weisen darauf hin, dass das Umfeldmodell in der Simulation identisch mit dem Umfeldmodell, welches im realen Versuchsfahrzeug zum Einsatz kommt, sein soll. Der sechste Schritt der Validierungsmethode ist das Vergleichen der realen und synthetischen Daten mit Metriken. Anschließend wird im letzten Schritt eine Entscheidung getroffen, ob das implementierte Sensormodell den Anforderungen genügt.

Die Methodik, an die diese Masterthesis angelehnt ist, baut auf der vorgestellten Methodik von Schärman et al. auf. Durch den Einsatz eines Lidar-Sensormodells kommt es zu einigen Änderungen der ursprünglichen Methodik, die im Folgenden beschrieben werden. Da sich die Sensorsimulation auf einen einzigen Sensortypen (Lidar) beschränkt, fällt der Schritt der HLF weg. Des Weiteren erfolgt die Ausgabe der beiden Lidar-Sensoren, die am Versuchsfahrzeug verbaut sind, nur im Verbund. Die Ausgabe der Daten erfolgt als (von der ECU) fusionierte Punktwolke. Aus diesem Grund fallen der LLF-Schritt und damit das gesamte Umfeldmodell weg.

Die Methodik wird auf eine Tool Chain übertragen (siehe Abbildung 4-4), die die Grundlage für die Datenbeschaffung der realen und synthetischen Daten sowie des quantitativen Bewertungsschrittes bildet. Messdaten werden bei definierten Szenarien mit dem realen Versuchsfahrzeug aufgezeichnet. GPS- sowie Delta-Daten zu anderen Objekten werden mittels eines hochpräzisen Kreismesssystems ADMA (*Automotive Dynamic Motion Analyzer*) ermittelt. Diese Daten werden für die Erstellung der Szenarien in VTD benötigt. Parallel zur Aufzeichnung der ADMA-Daten erfolgt die Aufzeichnung

⁶⁰ Vgl. Aeberhard, M.: Dissertation, Object-Level Fusion for Surround Environment Perception (2017), S. 28–32.

⁶¹ Mit Änderungen entnommen aus Schaermann, A. et al.: Validation of Vehicle Environment Sensor Models (2017), S. 408.

der Lidar-Sensordaten mit der Software ILV Premium. Die erzeugten idc-Dateien beinhalten neben den Punktwolken auch die Objektliste. Nach der Erstellung der Szenarien in VTD folgen die Simulation und die parallele Datenaufzeichnung als bag-Dateien. Diese bag-Dateien beinhalten die synthetisch generierte Punktwolke und werden den Auswerteargorithmen, die im ROS-Framework implementiert sind, für die weitere Verarbeitung zur Verfügung gestellt. Im letzten Schritt erfolgt die Bewertung der Simulation mit Metriken in Matlab, für die neben den idc- und bag-Dateien noch die Objektlisten als csv-Datei herangezogen werden.

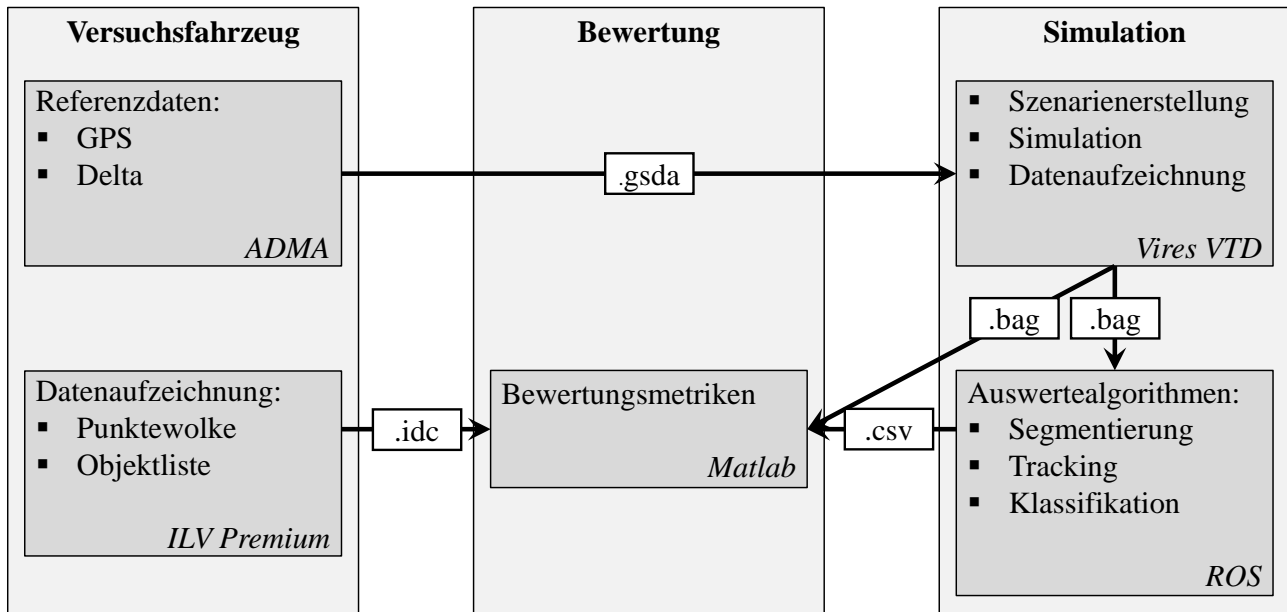


Abbildung 4-4: Tool Chain zur Bewertung der Lidar-Sensorsimulation (in Anlehnung an Schärmann et al.⁶²)

⁶² Vgl. Schaermann, A. et al.: Validation of Vehicle Environment Sensor Models (2017), S. 409.

5 Metriken zur Bewertung der Verarbeitungsebenen

In diesem Kapitel werden alle gefundenen Metriken, die in der Literatur gelistet sind und die für die Bewertung einzelner oder mehrerer Verarbeitungsebenen angewendet werden, vorgestellt. Das Kapitel ist untergliedert in die einzelnen Verarbeitungsebenen, denen die Metriken jeweils zugeordnet werden. Vor der Vorstellung der Metriken erfolgt zunächst die Definition und Abgrenzung der einzelnen Verarbeitungsebenen. Dieses Kapitel wird abgeschlossen mit der Vorstellung der Überlegungen zur Erstellung eines Gesamtmaßes sowie zur Metrikenauswahl.

5.1 Definition und Abgrenzung der Verarbeitungsebenen

Die Metriken für die Bewertung der Lidar-Sensorsimulation werden in vier verschiedene Verarbeitungsebenen eingeteilt. Jede Verarbeitungsebene besitzt dabei unterschiedliche Ausgänge, die wiederum als Eingang der Metriken dienen. Es werden vier Verarbeitungsebenen definiert, die sich am Aufbau der Lidar-Sensorsimulation (vgl. Abschnitt 2.3) orientieren.

Die erste Verarbeitungsebene ist die Erfassung der Umgebung mit den Lidar-Sensoren. Die Ausgabe dieser Verarbeitungsebene ist die reale oder simulierte Punktwolke. Daran anschließend folgt die Segmentierung, die die zweite Verarbeitungsebene darstellt. Ausgabe der Segmentierung ist die segmentierte Punktwolke. In dieser Punktwolke hat jeder Punkt eine feste Zugehörigkeit zu einem Segment. Das Tracking ist die dritte Verarbeitungsebene. Den Segmenten wird in diesem Schritt eine ID zugeordnet, die sie während ihrer gesamten Lifetime beibehalten. Da das Tracking über mehrere Scans erfolgt, wird neben der ID und der Lifetime auch der Track, also die Position des Objekts in den einzelnen Scans, ausgegeben. Außerdem werden beim Tracking die Abmessungen sowie Geschwindigkeiten der Objekte geschätzt. Die letzte Verarbeitungsebene ist die Klassifikation, deren Ausgang eine Objektliste darstellt, bei der jedem Objekt eine Klasse individuell zugeteilt ist.

5.2 Punktwolken

Dieser Abschnitt befasst sich mit Metriken, die zum Vergleich zweier Punktwolken angewendet werden. Eine Bewertung zur Anwendbarkeit der einzelnen Metriken für den Vergleich der Lidar-Sensorsimulation mit den realen Messdaten erfolgt in Kapitel 6. In Tabelle 5-2 am Ende dieses Abschnitts befindet sich eine Zusammenfassung aller vorgestellten Metriken für Punktwolken.

Schärmann et al.⁶³ stellen in ihrer Veröffentlichung drei Metriken vor, die einen Vergleich von Punktwolken zulassen. Die Metriken beschränken sich nicht ausschließlich auf Punktwolken und werden auch für den Vergleich von Occupancy Grids (OG) herangezogen. Die Autoren erstellen hierzu aus den Punktwolken zweidimensionale Occupancy Grids, bei denen jede Zelle eine der folgenden drei Zustände einnimmt: frei, besetzt oder undefiniert, falls die Zelle nicht vom Sensor gesehen wird. Für die erste Metrik greifen Schärmann et al. auf den von Carlson et al. definierten *OverallError* zurück. Für dessen Berechnung wird für jede Zelle der realen und simulierten Occupancy Grids

⁶³ Vgl. Schaermann, A. et al.: Validation of Vehicle Environment Sensor Models (2017), S. 409.

(OG_R bzw. OG_S) die Differenz der Belegungswerte ermittelt. Der Belegungswert einer Zelle nimmt einen Wert zwischen Null und Eins an und beträgt genau 0,0 bzw. 1,0, wenn die Zelle frei bzw. belegt ist. Die Summe aller absoluten Differenzwerte bildet den *OverallError*⁶⁴ zu

$$OverallError = \sum_{x=0}^{Breite} \sum_{y=0}^{Höhe} |OG_S(x, y) - OG_R(x, y)|. \quad (5-1)$$

Die Summationsindizes x und y stehen dabei für die Ausdehnung der Occupancy Grids in den beiden Raumrichtungen. Je niedriger der *OverallError* ist, desto größer ist die Übereinstimmung von realem und simuliertem Occupancy Grid. Bei vollständiger Deckungsgleichheit beider Occupancy Grids ist der *OverallError* = 0. Auch Pieringer⁶⁵ verwendet die *OverallError*-Metrik für die Evaluierung von Occupancy Grids. Schärman et al. berechnen den *OverallError* gleichermaßen für Punktwolken. O'Sullivan führt einen *Match Score* ein, dessen Berechnung der des *OverallErrors* folgt, mit dem Unterschied, dass statt dem Betrag der Differenzen die Differenzen quadriert werden, um nur positive Werte zu summieren.^{66a}

$$Match\ Score = \sum_{x=0}^{Breite} \sum_{y=0}^{Höhe} [OG_S(x, y) - OG_R(x, y)]^2 \quad (5-2)$$

Neben dem *OverallError* berechnen Schärman et al. zwei Korrelationskoeffizienten, den Kreuzkorrelationskoeffizienten nach Baron sowie den Korrelationskoeffizienten nach Pearson. Der Kreuzkorrelationskoeffizient nach Baron⁶⁷, den auch O'Sullivan^{66b} und Fernández-Maridgal und Blanco Claraco⁶⁸ zum Vergleich zweier Occupancy Grids verwenden, berechnet sich zu

$$C_B = \frac{\langle OG_S \cdot OG_R \rangle - \langle OG_S \rangle \langle OG_R \rangle}{\sigma(OG_S) \sigma(OG_R)} \quad (5-3)$$

mit dem Durchschnittsoperator $\langle \rangle$, der den Durchschnittswert aller freien und belegten Zellen berechnet sowie der Standardabweichung σ der Occupancy Grids. Das Ergebnis des Kreuzkorrelationskoeffizienten nach Baron liegt im Bereich zwischen minus eins und eins für keine bzw. vollständige Übereinstimmung der Occupancy Grids.

Der Korrelationskoeffizient nach Pearson C_P mit einem Wertebereich zwischen -1,0 und 1,0 (-1,0 bzw. 1,0 für vollständig negative bzw. positive Korrelation, 0,0 für keine Korrelation⁶⁹) ist in Guyon et al.⁷⁰ beschrieben. Er ist der Quotient der Kreuzkovarianz und den Standardabweichungen der beiden Occupancy Grids.

⁶⁴ Carlson, J. et al.: Conflict Metric as a Measure of Sensing Quality (2005), S. 2036.

⁶⁵ Vgl. Pieringer, C.: Masterarbeit, Modellierung des Fahrzeugumfelds mit Occupancy Grids (2013), S. 86.

⁶⁶ Vgl. O'Sullivan, S.: Master's Thesis, Empirical Evaluation Of Map Building Methodologies (2003), a: S. 166; b: S. 164.

⁶⁷ Vgl. Baron, R. J.: Mechanisms of human facial recognition (1981).

⁶⁸ Vgl. Fernández-Madriral, J.-A.; Blanco Claraco, J. L.: Simultaneous localization and mapping (2013), S. 195–196.

⁶⁹ Vgl. Carlson, J. et al.: Conflict Metric as a Measure of Sensing Quality (2005), S. 2036.

⁷⁰ Guyon, I. et al.: Feature Extraction (2006), S. 9.

$$C_p = \frac{\sum_{i=1}^m (x_i - \bar{x}) (y_i - \bar{y})}{\sqrt{\sum_{i=1}^m (x_i - \bar{x})^2 \sum_{i=1}^m (y_i - \bar{y})^2}} \quad (5-4)$$

Dabei sind x_i und y_i die m -dimensionalen Vektoren, die die Werte der simulierten bzw. realen Occupancy Grids enthalten, und \bar{x} bzw. \bar{y} die Mittelwerte der Vektoren über den Index i . Schärmann et al. berechnen die beiden Koeffizienten sowohl für Occupancy Grids als auch für Punktwolken. Neben den hier aufgelisteten Metriken vergleichen die Autoren jeweils die Punktwolken und Occupancy Grids visuell. Außerdem überprüfen sie die Empfindlichkeit der Metriken, indem das virtuelle Szenario modifiziert wird, während das reale Szenario nicht geändert wird. Die Ergebnisse zeigen, dass die Metriken darauf ansprechen und sich verschlechtern.

Hanke et al. ergänzen zu den drei Metriken von Schärmann et al., dass eine Anwendung dieser auf die Punktwolken nur möglich sei, wenn vor der Berechnung der Metriken eine Euklidische Abstands-Matrix (EDM, euclidean distance matrix) erstellt wird.⁷¹ Die von Hanke et al. berechnete $m \times n$ EDM beinhaltet den Abstand aller Punkte zum Sensor. Die Parameter m und n sind abhängig von der Anzahl der Kanäle bzw. von der horizontalen Winkelauflösung und dem horizontalen Öffnungswinkel. Als Erweiterung der Occupancy Grids verwenden Hanke et al. sogenannte Scan Grids (SG). SGs sind OGs, die nur die Daten eines einzelnen Scans beinhalten. Mehrere über die Zeit akkumulierte SGs bilden ein OG.⁷¹

Grewe et al.^{72a} schreiben, dass die Bewertung eines Occupancy Grids häufig nur visuell stattfindet, was zu subjektiven Ergebnissen führt. Daher schlagen sie die drei Metriken *Map Score*, *Occupied Cells Ratio* und *Free Cells Ratio* vor. Die ursprüngliche Definition des *Map Scores* stammt von Martin und Moravec⁷³ und berechnet sich zu

$$Map\ Score = \sum_i [1 + \log_2(OG_{R,i}OG_{S,i} + \overline{OG}_{R,i}\overline{OG}_{S,i})] \quad (5-5)$$

mit dem Laufindex i über alle Zellen des realen und simulierten Occupancy Grids und der Wahrscheinlichkeit $OG_{R,i}OG_{S,i}$, dass beide Zellen belegt sind und der Gegenwahrscheinlichkeit $\overline{OG}_{R,i}\overline{OG}_{S,i}$, dass beide Zellen frei sind. Der *Map Score* konvergiert dabei gegen die Anzahl der Zellen, je ähnlicher die beiden Occupancy Grids werden. Bei vollständiger Deckungsgleichheit entspricht der *Map Score* der Anzahl der Zellen.

Bei den beiden Verhältnissen *Occupied* und *Free Cells Ratio* wird das zu testende Occupancy Grid mit dem Referenz-Occupancy Grid verglichen. Das *Occupied Cells Ratio* ist dabei das Verhältnis zwischen den als richtig klassifizierten belegten Zellen und der gesamten Anzahl an belegten Zellen im Referenz-Occupancy Grid. Analog dazu wird das *Free Cells Ratio* berechnet.⁷⁴

⁷¹ Vgl. Hanke, T. et al.: Validation of Virtual Point Cloud Data (2017), S. 5.

⁷² Vgl. Grewe, R. et al.: Accuracy of an Automotive Occupancy Grid (2012), a: -;b: S. 19.

⁷³ Martin, M. C.; Moravec, H. P.: Robot Evidence Grids (1996), S. 6.

⁷⁴ Grewe, R. et al.: Accuracy of an Automotive Occupancy Grid (2012), S. 21.

$$\text{Occupied Cells Ratio} = \frac{\sum \text{cells}_{\text{test map,occ,true}}}{\sum \text{cells}_{\text{reference map,occ}}} \quad (5-6)$$

$$\text{Free Cells Ratio} = \frac{\sum \text{cells}_{\text{test map,free,true}}}{\sum \text{cells}_{\text{reference map,free}}} \quad (5-7)$$

Grewe et al. verweisen auf den Unterschied zwischen dem *Map Score* und den beiden Verhältnissen. Während die Berechnung des *Map Scores* auch bei probabilistischen Zellenwerten (Werte im Bereich zwischen Null und Eins) möglich ist, können für die beiden Verhältnisse jeweils nur binäre Zellenwerte (frei = 0 und belegt = 1) berücksichtigt werden. Des Weiteren betonen die Autoren, dass bereits bei der Erstellung der Occupancy Grids der Kompromiss zwischen der Zellengröße (Rechenaufwand) und der Genauigkeit (Diskretisierungsfehler) gefunden werden muss.^{72b,75}

Für die Gegenüberstellung verschiedener Simulationsmodelle schlagen Browning et al. mehrere Metriken vor, die im Folgenden dargestellt werden. Die erste Metrik ist der Vergleich von Entfernungshistogrammen, deren Anwendung nur bei statischen Szenarien möglich ist. Aus den realen und simulierten Punktwolken werden Histogramme über die Entfernung oder Wahrscheinlichkeitsdichtefunktionen (pdf, probability density function) berechnet, die anschließend mit der Bhattacharyya-Entfernungsmetrik verglichen werden. Gegeben seien zwei normierte Histogramme $H = (h_1, \dots, h_N)$ und $F = (f_1, \dots, f_N)$ mit jeweils N Klassen und $\sum_i h_i = 1$, $\sum_i f_i = 1$, dann folgt für die Bhattacharyya-Entfernung^{76a}

$$D_B(H, F) = -\log \sum_i \sqrt{h_i f_i}. \quad (5-8)$$

Für zwei identische Histogramme H und F ist die Bhattacharyya-Entfernung $D_B = 0$. Je geringer die Überdeckung der Histogramme ist, desto größer ist die Bhattacharyya-Entfernung. Im Falle keiner Überdeckung der Histogramme strebt D_B gegen $+\infty$ ($h_i > 0 \rightarrow f_i = 0$ und $f_i > 0 \rightarrow h_i = 0$).

Die zweite Metrik, die bei Browning et al. zur Anwendung kommt, ist die Punktwolke-Distanz-Metrik (*Point Cloud Distance Metric*). Bei dieser wird die minimale Distanz aller Punkte einer Punktwolke $X = (x_1, \dots, x_M)$ zur anderen Punktwolke $Y = (y_1, \dots, y_N)$ mit $x_i, y_j \in \mathbb{R}^3$ berechnet. Die Autoren definieren die Punktwolke-zu-Punktwolke-Distanz zu

$$D'_{PP}(X, Y) = \sum_i \min_j \|x_i - y_j\| \quad (5-9)$$

und da dies eine unsymmetrische Metrik ist, wird der ungünstigste Fall ausgewählt

$$D_{PP}(X, Y) = \max(D'_{PP}(X, Y), D'_{PP}(Y, X)). \quad (5-10)$$

Berechnet wird die Punktwolke-zu-Punktwolke-Distanz sowohl für Punktwolken, die bei einem einzelnen Scan aufgezeichnet werden, als auch für Punktwolken, die aus mehreren Scan bestehen.

⁷⁵ Vgl. Burgard, W.; Hebert, M.: World Modeling (2008), S. 855.

⁷⁶ Browning, B. et al.: Unmanned Ground Vehicle Lidar Simulation (2012), a: S. 1361; b: S. 1362.

Mittels einer weiteren Metrik untersuchen Browning et al. das Verhalten der Simulation bei sogenannten *no-returns*, also den Fällen, bei denen kein Echo detektiert wird. Insgesamt gibt es vier mögliche Treffer-Fälle, die in Tabelle 5-1 dargestellt sind. Es sind hohe *Hit*- bzw. *Miss-detected*-Raten anzustreben. Eine geringe *Hit-detected*-Rate weist auf ein nicht vollständiges Modell hin, während eine geringe *Miss-detected*-Rate signalisiert, dass Teile des Modells auf freien Raum übergreifen.⁷⁷

Tabelle 5-1: Hit/Miss-Vergleich nach Browning et al.^{76b}

		Real Data	
		Return	No-return
Simulated Data	Return	Hit detected	False hit
	No-return	False miss	Miss detected

Lee beschreibt die Eigenschaften, die eine Metrik für das Bewerten einer Karte (Occupancy Grid) haben muss, wie folgt:⁷⁸

- Die Metrik muss eindeutig definiert sein.
- Die Metrik muss mehrwertig sein.
- Die Metrik muss den Zweck der Karte widerspiegeln.
- Die Metrik muss einen Kompromiss zwischen Abdeckung und Detail finden.
- Die Metrik muss während der Konstruktion der Karte anwendbar sein.

Collins et al.⁷⁹ leiten nach diesen Eigenschaften mehrere Metriken ab. Einer diese Metriken ist der Kreuzkorrelationskoeffizient nach Baron, den auch Schärmann et al. berechnen (vgl. Formel (5-3)). Die Berechnung des *MapScores* folgt der Berechnung des *Match Scores* von O’Sullivan (Gleichung (5-2)). Collins et al. schlagen eine Erweiterung des *MapScores* vor, da das Ergebnis des *MapScores* bei verschiedenen Szenarien aufgrund fehlender Normierung nicht verglichen werden kann. Für die Normierung wird das OG mit den schlechtesten möglichen Zellwerten berechnet. Hierfür wird freien Zellen der Wert 1,0 (belegt) und belegten Zellen der Wert 0,0 (frei) zugewiesen. Der Wert der belegten Zellen wird jedoch nur von 1,0 auf 0,0 verändert, wenn sich die Zelle innerhalb der maximalen Distanz des Sensors befindet.⁷⁹ Dieses Verfahren erfordert jedoch Kenntnis über die Ground-Truth-Karte. Eine zusätzliche Erweiterung des *MapScores* ist das Berücksichtigen von ausschließlich belegten Zellen (Zellwert > 0,5) in der Berechnung. Collins et al. begründen dies damit,

⁷⁷ Vgl. Browning, B. et al.: Unmanned Ground Vehicle Lidar Simulation (2012), S. 1362.

⁷⁸ Lee, D.: Map-building and exploration strategies (1996), S. 106.

⁷⁹ Vgl. Collins, T. et al.: Occupancy Grid Mapping (2007), S. 3–4.

dass bei OGs meistens mehr freie als belegte Zellen vorhanden sind und das Erkennen von Hindernissen (belegten Zellen) korrekt funktionieren muss.⁷⁹

Balaguer et al.⁸⁰ verwenden neben den bereits vorgestellten Korrelationskoeffizienten (siehe Baron Formel (5-3) und Pearson Formel (5-4)) die *Picture-Distance-Function* (PDF). Die ursprüngliche Definition der PDF stammt von Birk⁸¹ und dient dem Vergleich von Bildern. Bei der PDF wird für jede belegte Zelle eines OGs die Manhattan-Distanz zur nächstgelegenen belegten Zelle des Referenz-OGs berechnet. Auf gleiche Weise wird für jede belegte Zelle des Referenz-OGs die Manhattan-Distanz zur nächstgelegenen Zelle des zu vergleichenden OGs berechnet. Die Summe aller Distanzen geteilt durch die Anzahl aller belegten Zellen (von Referenz-OG und zu vergleichendem OG) ergibt das Ergebnis der PDF. Bei vollständiger Übereinstimmung der beiden OGs ist PDF = 0, die PDF ist jedoch nicht normiert. Die OPDF (*Occupied Picture-Distance-Function*) von Baizid et al. basiert auf der PDF und ist im Intervall [0,1] definiert zu⁸²

$$OPDF = \left(1 - \frac{\sum_{i=1}^{no} d_i}{no \cdot r}\right). \quad (5-11)$$

Darin steht *no* für die Anzahl der belegten Zellen, d_i für die Manhattan-Distanz und $r = \sqrt{width^2 + height^2}$ für einen maximalen Suchradius mit den Parametern *width* und *height*. Analog zu der OPDF wird die UPDF (*Unoccupied Picture-Distance-Function*) definiert, deren Berechnung sich auf freie Zellen (Gesamtanzahl *nu*) beschränkt. Sowohl OPDF als auch OPDF sind normiert und nehmen bei vollständiger Übereinstimmung der OGs den Wert Eins an

$$UPDF = \left(1 - \frac{\sum_{i=1}^{nu} d_i}{nu \cdot r}\right). \quad (5-12)$$

Eine erweiterte Berechnung aller vorgestellten Metriken kann erfolgen, indem zunächst die erstellten simulierten Occupancy Grids bearbeitet werden. Mögliche Bearbeitungsschritte sind das Verschieben, Drehen oder Skalieren der Occupancy Grids. Nach jedem Bearbeitungsschritt wird zwischen den realen und den bearbeiteten simulierten Occupancy Grids der Kreuzkorrelationskoeffizient berechnet, wobei dessen Maximierung angestrebt wird. Für das simulierte Occupancy Grid mit maximalem Kreuzkorrelationskoeffizienten werden anschließend alle Metriken berechnet. Bei der Berechnung der Metriken kann außerdem berücksichtigt werden, wie viele Bearbeitungsschritte für die Maximierung des Kreuzkorrelationskoeffizienten benötigt werden.

Tabelle 5-2: Übersicht Metriken für Punktwolken

Metrik	Formel	Verwendung in
Overall Error	$OverallError = \sum_{x=0}^{Breite} \sum_{y=0}^{Höhe} OG_S(x, y) - OG_R(x, y) $	Schärmann et al., Hanke et al., Carlson et al., Balaguer et al., Pieriger

⁸⁰ Vgl. Balaguer, B. et al.: Evaluating maps from RoboCup (2009).

⁸¹ Vgl. Birk, A.: Learning Geometric Concepts with an Evolutionary Algorithm, S. 3.

⁸² Baizid, K. et al.: Vector Maps: A Lightweight and Accurate Map Format for Multi-robot Systems (2016), S. 424.

Metrik	Formel	Verwendung in
Match Score	$Match\ Score = \sum_{x=0}^{Breite} \sum_{y=0}^{Höhe} [OG_S(x, y) - OG_R(x, y)]^2$	O'Sullivan, Balaguer et al., Collins et al. (nur für belegte Zellen)
Kreuzkorrelationskoeffizient nach Baron	$C_B = \frac{\langle OG_S \cdot OG_R \rangle - \langle OG_S \rangle \langle OG_R \rangle}{\sigma(OG_S) \sigma(OG_R)}$	Schärmann et al., Hanke et al., Collins et al., Balaguer et al., Baron, Fernández-Maridgal und Blanco Claraco, Baizid et al.
Korrelationskoeffizient nach Pearson	$C_P = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^m (x_i - \bar{x})^2 \sum_{i=1}^m (y_i - \bar{y})^2}}$	Schärmann et al., Hanke et al., Balaguer et al., Guyon et al., Baizid et al.
Map Score	$Map\ Score = \sum_i [1 + \log_2(OG_{R,i} OG_{S,i} + \overline{OG}_{R,i} \overline{OG}_{S,i})]$	Grewe et al., Martin und Moravec, Fernández-Maridgal und Blanco Claraco
Map Score nur für belegte Zellen	$Map\ Score = \sum_i [1 + \log_2(OG_{R,i} OG_{S,i} + \overline{OG}_{R,i} \overline{OG}_{S,i})]$	Grewe et al., Balaguer et al., O'Sullivan
Occupied Cells Ratio	$Occupied\ Cells\ Ratio = \frac{\sum cells_{test\ map, occ, true}}{\sum cells_{reference\ map, occ}}$	Grewe et al.
Free Cells Ratio	$Free\ Cells\ Ratio = \frac{\sum cells_{test\ map, free, true}}{\sum cells_{reference\ map, free}}$	Grewe et al.
Bhattacharyya-Entfernung	$D_B(H, F) = -\log \sum_i \sqrt{h_i f_i}$	Browning et al.
Punktewolke-zu-Punktewolke-Distanz	$D_{PP}(X, Y) = \max(D'_{PP}(X, Y), D'_{PP}(Y, X)) \text{ mit}$ $D'_{PP}(X, Y) = \sum_i \min_j \ x_i - y_j\ $	Browning et al.
Hit/Miss-Vergleich	$Hit\ detected\ rate = \frac{Hit\ detected}{Returns + No-returns}$ $Miss\ detected\ rate = \frac{Miss\ detected}{Returns + No-returns}$	Browning et al.
Occupied Picture-Distance-Function	$OPDF = \left(1 - \frac{\sum_{i=1}^{no} d_i}{no \cdot r}\right)$	Baloguer et al., Baizid et al., Birk
Unoccupied Picture-Distance-Function	$UPDF = \left(1 - \frac{\sum_{i=1}^{nu} d_i}{nu \cdot r}\right)$	Baizid et al.

5.3 Segmentierung

In diesem Abschnitt werden Metriken vorgestellt, die eine Bewertung des Segmentierungsmoduls ermöglichen. Da speziell für die Bewertung der Segmentierung von Sensormodellen wenige Metriken existieren, wird auf Metriken für den Vergleich von Segmentierungsalgorithmen zurückgegriffen.

Douillard et al.⁸³ vergleichen mehrere Segmentierungsalgorithmen und evaluieren deren Leistung mittels zweier Metriken. Der Vergleich der von dem Algorithmus gefundenen Segmente (siehe Beispiel A in Abbildung 5-1) erfolgt dabei immer mit einem manuell segmentierten Datensatz (siehe Beispiel B in Abbildung 5-1). Bei der Metrik *Point Score Range* wird dem größten Segment eines Objekts in A der Status Treffer zugeteilt (hier der obere Teil der Baumkrone). Die anderen Segmente werden als Fehler gewertet. Ein Segment erhält dabei nur einmal den Status Treffer. Der Prozess läuft vom größten manuell segmentierten Objekt zum kleinsten. Falls ein vom Algorithmus gebildetes Segment in mehreren manuell segmentierten Objekten auftaucht, so bekommt es nur im größten manuellen segmentierten Objekt den Status Treffer und in allen anderen Segmenten den Status Fehler. Nachdem allen Segmenten ein Status zugeordnet ist, wird das Verhältnis zwischen Treffern und Fehlern aller Punkte innerhalb der Segmente berechnet.

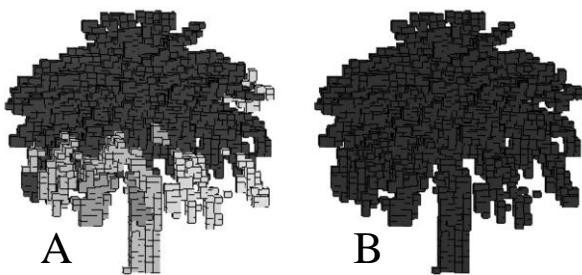


Abbildung 5-1: Vergleich der Segmentierung des Segmentierungsalgorithmus (A) und manueller Segmentierung (B)⁸⁴

Eine weitere Metrik, die auf der gleichen Idee basiert, ist die *Voxel Score Range*. Der Unterschied zur *Point Score Range* ist, dass nicht einzelne Punkte als Treffer oder Fehler markiert werden, sondern Voxel. Diese Quader (siehe Abbildung 5-1) beinhalten mehrere Punkte. Der Vorteil dieser Methode liegt nach Douillard et al. bei der Anwendung der Metriken bei Punktwolken mit geringer Dichte, da bei diesen ein Großteil der gesamten Punkteanzahl aus Bodenreflexionen besteht. Die Voxelanzahl mit Bodenreflexionen ist dagegen im Datensatz von Douillard et al. im Verhältnis zur gesamten Voxelanzahl deutlich geringer.⁸³

Shin et al.⁸⁵ präsentieren mehrere Metriken für den Vergleich ihres entwickelten Segmentierungsalgorithmus mit anderen Algorithmen. Hierzu definieren die Autoren *target objects*. Dies sind alle Objekte (Fahrzeuge, Fußgänger, Bäume etc., ausgenommen Häuser und Büsche) im Umkreis von 70

⁸³ Vgl. Douillard, B. et al.: Segmentation of 3D LIDAR point clouds (2011), S. 2802.

⁸⁴ Mit Änderungen entnommen aus Douillard, B. et al.: Segmentation of 3D LIDAR point clouds (2011), S. 2802.

⁸⁵ Vgl. Shin, M.-O. et al.: Segmentation of 3-D Point Clouds (2017), S. 3371.

Metern vom Lidar-Sensor und mit mehr als 30 Punkten. Jedem *target object* wird ein Status zugeordnet: True Positive (TP), False Positive (FP) oder False Negative (FN). Deren Definition nach Shin et al. lautet:

- TP: *Target objects*, die vom Algorithmus erkannt werden und erfolgreich von anderen Objekten separiert werden.
- FN: *Target objects*, die nicht vom Algorithmus erkannt werden oder zu weniger als 2/3 segmentiert werden.
- FP: Sind keine *target objects*, sondern fälschlicherweise vom Algorithmus als *target object* identifiziert, z. B. Bodenreflexionen.

Basierend auf diesen Definitionen sind fünf Metriken von Shin et al. definiert⁸⁶:

$$OSR = \frac{TPs}{TPs + \text{over-segmentation}} \quad (5-13)$$

$$USR = \frac{TPs}{TPs + \text{under-segmentation}} \quad (5-14)$$

$$\text{Precision} = \frac{TPs}{TPs + FPs} \quad (5-15)$$

$$e\text{-Precision} = \frac{TPs}{TPs + FPs + \text{over-segmentation}} \quad (5-16)$$

$$\text{Recall} = \frac{TPs}{TPs + FNs} \quad (5-17)$$

OSR und *USR* sind die *Over-segmentation Suppression Rate* bzw. *Under-segmentation Suppression Rate* mit der Anzahl der über- und untersegmentierten *target objects* (*over-segmentation* und *under-segmentation*). Die effektive Präzision *e-Precision* wird eingeführt, um den Einfluss der Übersegmentierung auf die Präzision zu quantifizieren.⁸⁵

5.4 Tracking

Wie bei der Segmentierung gibt es für die Evaluierung der Qualität von Trackingalgorithmen einige Metriken. Diese werden in diesem Abschnitt vorgestellt. Die Überprüfung der Anwendbarkeit der Metriken für die Bewertung des Trackingmodells der Lidar-Sensorsimulation erfolgt in Kapitel 6.

Ackermann⁸⁷ stellt zur Leistungsbewertung von Trackingalgorithmen mehrere Metriken vor, darunter die Hausdorff-Metrik, die *Optimal-Mass-Transfer-Metrik* (OMAT-Metrik), die *Optimal-Subpattern-Assignment-Metrik* (OSPA-Metrik) sowie deren Erweiterungen *Optimal-Subpattern-Assignment-for-Tracks-Metrik* (OSPA-T-Metrik) und *Generalized-Optimal-Subpattern-Assignment-Metrik*

⁸⁶ Shin, M.-O. et al.: Segmentation of 3-D Point Clouds (2017), S. 3371.

⁸⁷ Vgl. Ackermann, S. M.: Masterthesis, Systematische Untersuchung von Radar Tracking-Algorithmen (2017), S. 48–58.

(GOSPA-Metrik). Außerdem wird eine Metrik von Rahmathullah et al. vorgestellt. Diese Metriken werden im Folgenden erläutert.

5.4.1 Hausdorff-Metrik

Die Hausdorff-Distanz $d^H(S, T)$ ist definiert als Abstand zwischen zwei nichtleeren kompakten Teilmengen S und T eines metrischen Raums mit einer Metrik $d(\mathbf{x}, \mathbf{y})$ (z. B. euklidischer Abstand) auf diesem Raum⁸⁸

$$d^H(S, T) = \max \left\{ \max_{\mathbf{x} \in S} \min_{\mathbf{y} \in T} d(\mathbf{x}, \mathbf{y}), \max_{\mathbf{y} \in T} \min_{\mathbf{x} \in S} d(\mathbf{x}, \mathbf{y}) \right\}. \quad (5-18)$$

Die Hausdorff-Metrik hat nach Hoffman und Mahler einen Nachteil bei der Berücksichtigung von Kardinalitätsfehlern, also einer unterschiedlichen Objektanzahl in den Teilmengen. Auf diese Art von Fehler reagiert die Hausdorff-Metrik nur in geringem Umfang.^{89a} Außerdem werden Track-Kontinuitätskriterien nicht beachtet und der Einfluss einzelner Messausreißer auf das Gesamtergebnis der Hausdorff-Metrik ist groß.⁹⁰

5.4.2 OMAT-Metrik

Hoffman und Mahler^{89b} entwickeln die *Optimal-Mass-Transfer*-Metrik (OMAT-Metrik) mit dem Ziel, die Nachteile der Hausdorff-Metrik zu beseitigen. Die OMAT-Metrik d_p mit der Ordnung p , wobei $1 \leq p < \infty$, und den finiten nichtleeren Teilmengen $X = \{x_1, \dots, x_m\}$ und $Y = \{y_1, \dots, y_n\}$ basiert auf der Wasserstein-Distanz und ist definiert zu

$$d_p(X, Y) := \min_C \left(\sum_{i=1}^m \sum_{j=1}^n C_{i,j} d(x_i, y_j)^p \right)^{1/p} \quad (5-19)$$

und

$$d_\infty(X, Y) := \min_C \max_{1 \leq i \leq m, 1 \leq j \leq n} \tilde{C}_{i,j} d(x_i, y_j). \quad (5-20)$$

Dabei stellen C die $m \times n$ Transportmatrizen $C = (C_{i,j})$ dar, mit $\tilde{C}_{i,j} = 1$ wenn $C_{i,j} \neq 0$ und $\tilde{C}_{i,j} = 0$ sonst. Eine Matrix ist eine Transportmatrix, wenn folgende Bedingungen gelten

$$\sum_{j=1}^n C_{i,j} = \frac{1}{m} \text{ für } 1 \leq i \leq m \text{ und } \sum_{i=1}^m C_{i,j} = \frac{1}{n} \text{ für } 1 \leq j \leq n. \quad (5-21)$$

⁸⁸ Hoffman, J. R.; Mahler, R.: Multitarget Miss Distance via Optimal Assignment (2004), S. 329.

⁸⁹ Vgl. Hoffman, J. R.; Mahler, R.: Multitarget Miss Distance via Optimal Assignment (2004), a: S. 329, b: -; c: S. 330.

⁹⁰ Vgl. Ackermann, S. M.: Masterthesis, Systematische Untersuchung von Radar Tracking-Algorithmen (2017), S. 48.

Mit der Ordnung p der OMAT-Metrik lässt sich die Gewichtung von Ausreißern festlegen. Je höher die Ordnung, desto stärker werden Ausreißer gewichtet, bis bei $d_\infty(X, Y)$ nur noch Ausreißer berücksichtigt werden. Mit steigender Ordnung bestraft die OMAT-Metrik also die Existenz von geschätzten Tracks, die im Ground Truth nicht vorhanden sind, immer stärker.^{89c}

Schuhmacher et al.⁹¹ beschreiben fünf Probleme bei der OMAT-Metrik, die anhand der Abbildung 5-2 erläutert werden:

- 1) Widersprüchlichkeit der Metrik: Ergebnis der OMAT-Metrik hängt davon ab, wie ausgeglichen die Anzahl zwischen geschätzten und Ground-Truth-Objekten ist (Szenario D bis F).
- 2) Künstliche Aufteilung der Schätzungen bei unterschiedlichen Kardinalitäten: Die OMAT-Metrik teilt die Schätzungen zwischen den Ground-Truth-Objekten auf, obwohl eine Zuordnung zum nächsten Objekt einfacher zu verstehen ist (Szenario E, die beiden linken Schätzungen werden anteilig dem rechten Ground-Truth-Objekt zugeordnet).
- 3) Abhängigkeit der Metrik von der räumlichen Verteilung der Objekte: Die OMAT-Metrik bestraft Kardinalitätsfehler stärker, je weiter die Objekte voneinander entfernt sind.
- 4) $d_p(X, Y)$ ist nicht definiert, wenn X oder Y leer sind (Szenario A).
- 5) Topologie lässt sich nicht etablierten Topologien (stochastische Geometrie oder Punktprozess-Theorie) zuordnen.

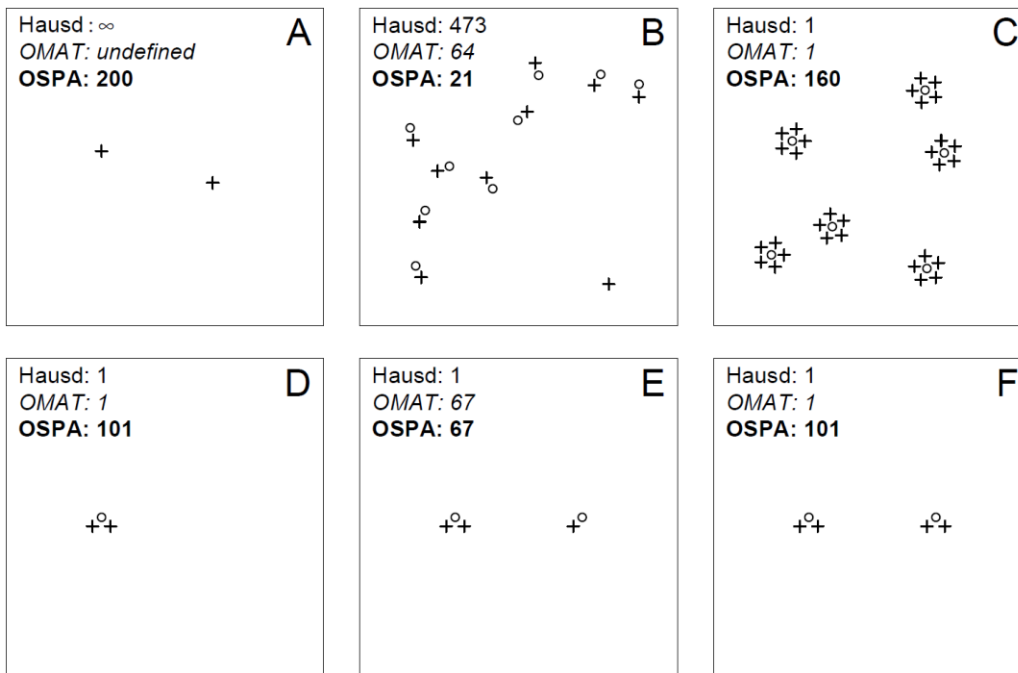


Abbildung 5-2: Sechs Szenarien zur Verdeutlichung der Stärken und Schwächen der Metriken. o: Ground-Truth-Objekte, +: geschätzte Objekte. Parameter $p = 1$ (für OMAT und OSPA) und $c = 200$ (für OSPA)⁹²

⁹¹ Vgl. Schuhmacher, D. et al.: Metric for Performance Evaluation of Multi-Object Filters (2008), S. 3449–3450.

⁹² Schuhmacher, D. et al.: Metric for Performance Evaluation of Multi-Object Filters (2008), S. 3448.

5.4.3 OSPA-Metrik

Schuhmacher et al. schlagen die *Optimal-Subpattern-Assignment*-Metrik (OSPA-Metrik) vor, die wie die OMAT-Metrik auf der Wasserstein-Distanz basiert, jedoch die Schwachstellen der OMAT-Metrik eliminiert. Die OSPA-Metrik $\bar{d}_p^{(c)}$ mit der Ordnung p und dem Cut-Off-Parameter c , wobei $1 \leq p < \infty$ und $c > 0$, ist für die beiden Teilmengen $X = \{x_1, \dots, x_m\}$ und $Y = \{y_1, \dots, y_n\}$ mit $m, n \in \mathbb{N}_0 = \{0, 1, 2, \dots\}$ definiert zu

$$\bar{d}_p^{(c)}(X, Y) := \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m d^{(c)}(x_i, y_{\pi(i)})^p + c^p(n-m) \right) \right)^{1/p} \quad (5-22)$$

wenn $m \leq n$, und $\bar{d}_p^{(c)}(X, Y) := \bar{d}_p^{(c)}(Y, X)$ wenn $m > n$, außerdem

$$\bar{d}_\infty^{(c)}(X, Y) := \begin{cases} \min_{\pi \in \Pi_n} \max_{1 \leq i \leq n} d^{(c)}(x_i, y_{\pi(i)}) & \text{wenn } m = n, \\ c & \text{wenn } m \neq n, \end{cases} \quad (5-23)$$

mit der Distanz $d^{(c)}(x, y) := \min(c, d(x, y))$ zwischen x und y und mit dem Cut-Off bei $c > 0$ sowie mit der Menge von Permutationen Π_k auf $\{1, 2, \dots, k\}$ für alle $k \in \mathbb{N} = \{1, 2, \dots\}$. Wenn $m = n = 0$, dann wird die Distanz auf Null gesetzt. Schuhmacher et al. zeigen eine Methode auf, die die Berechnung der OSPA-Metrik in drei Schritten ermöglicht, sofern $p < \infty$ und $m \leq n$:

- 1) Die optimale Zuordnung aller Objekte von X zu den nächstgelegenen Objekten in Y soll gefunden werden, bspw. mit der Ungarischen Methode⁹³.
- 2) Für alle n Objekte y_j in Y wird jeweils der Parameter α_j berechnet. Dieser entspricht entweder dem Cut-Off-Parameter c , falls dem Objekt y_j kein Objekt x_i zugeordnet wird, oder dem Minimum von c und der Distanz zum zugeordneten Objekt x_i .
- 3) Aus allen Parametern α_j wird die OSPA-Metrik durch Berechnung des Durchschnitts p -ter Ordnung $\left(\frac{1}{n} \sum_{j=1}^n \alpha_j^p \right)^{1/p}$ ermittelt.

5.4.4 OSPA-T-Metrik

Die OSPA-Metrik ist geeignet für die Bewertung von Multi-Object-Filtering-Systemen. Ristic et al.⁹⁴ passen die OSPA-Metrik an, damit diese eine Bewertung von Multi-Target-Trackingalgorithmen zulässt. Das Ziel der abgewandelten Metrik, *OSPA-for-Tracks* (OSPA-T), ist das Messen der Distanz zwischen der Ground-Truth-Trajektorie und der geschätzten Trajektorie des Trackingalgorithmus. Die Erweiterung der OSPA-Metrik berücksichtigt zusätzliche Kriterien wie z. B. die Track-Kontinuität. Für die Berechnung der OSPA-T-Metrik werden Trajektorien X der zeitlichen Länge K definiert,

⁹³ Vgl. Papadimitriou, C.; Steiglitz, K.: Combinatorial optimization (1998), S. 248–255.

⁹⁴ Vgl. Ristic, B. et al.: A Metric for Performance Evaluation of Multi-Target Tracking Algorithms (2011), S. 3453–3454.

deren Elemente X_k mit $k = 1, \dots, K$ zu den zeitdiskreten Werten $\tau = (t_1, t_2, \dots, t_K)$ entweder leer oder mit der Elementarmenge (ℓ, x_k) befüllt sind

$$X = (X_1, X_2, \dots, X_K). \quad (5-24)$$

Dabei stellt $\ell \in \mathbb{N}$ die Bezeichnung der Trajektorie dar, die zeitunabhängig ist, und x_k ist der Zustand der Trajektorie wie z. B. die Position und Geschwindigkeit in kartesischen Koordinaten. Es wird ein Indikator e_k eingeführt, der den Wert Eins annimmt, wenn die Trajektorie zum Zeitpunkt t_k existiert und sonst den Wert Null annimmt. Somit ist

$$X_k = \begin{cases} \emptyset, & \text{wenn } e_k = 0 \\ \{(\ell, x_k)\}, & \text{wenn } e_k = 1. \end{cases} \quad (5-25)$$

Wie bei der Berechnung der OSPA-Metrik werden bei der OSPA-T-Metrik zwei Teilmengen X und Y verglichen. Diese Teilmengen sind zum Zeitpunkt t_k

$$X_k = \{(\ell_1, x_{k,1}), \dots, (\ell_m, x_{k,m})\} \quad (5-26)$$

und

$$Y_k = \{(s_1, y_{k,1}), \dots, (s_n, y_{k,n})\}. \quad (5-27)$$

Die OSPA-T-Distanz zwischen den Teilmengen ist für $m \leq n$ analog zur OSPA-Distanz (vgl. Gleichung (5-22)) definiert zu

$$D_{p,c}(X_k, Y_k) = \left[\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m \left(d_c(\tilde{x}_{k,i}, \tilde{y}_{k,\pi(i)}) \right)^p + (n-m) \cdot c^p \right) \right]^{1/p}, \quad (5-28)$$

mit der Cut-Off-Distanz $d_c(\tilde{x}, \tilde{y}) = \min(c, d(\tilde{x}, \tilde{y}))$ zwischen zwei Trajektorien zum Zeitpunkt t_k und dem Cut-Off-Parameter c , wobei $c > 0$. Die Basisdistanz $d(\tilde{x}, \tilde{y})$ zwischen zwei Trajektorien zum Zeitpunkt t_k ist für $1 \leq p' < \infty$ wie folgt definiert

$$d(\tilde{x}, \tilde{y}) = (d(x, y)^{p'} + d(\ell, s)^{p'})^{1/p'}. \quad (5-29)$$

Darin ist $d(x, y)$ die Lokalisierungs-Basisdistanz (typischerweise $d(x, y) = \|x - y\|_{p'}$) und $d(\ell, s)$ der Benennungsfehler, wobei dieser durch

$$d(\ell, s) = \alpha \bar{\delta}[\ell, s] \quad (5-30)$$

definiert ist. Das Komplement des Kronecker-Deltas ist $\bar{\delta}[\ell, s] = 0$, wenn $\ell = s$ und $\bar{\delta}[\ell, s] = 1$ sonst. Der Parameter $\alpha \in [0, c]$ ist für die Gewichtung des Benennungsfehlers verantwortlich, wobei $\alpha = 0$ keine Gewichtung und $\alpha = c$ maximale Gewichtung bedeutet.

5.4.5 GOSPA-Metrik

Rahmahtullah et al.⁹⁵ entwickeln die *Generalized-OSPA*-Metrik (GOSPA-Metrik), die wie die OSPA-T-Metrik auf der OSPA-Metrik aufbaut. Ein Nachteil der OSPA-Metrik ist, dass Trackingalgorithmen, welche viele falsche und verfehlte Trajektorien erkennen, trotzdem geringe OSPA-Distanzen erzielen können. Der Ansatz der Autoren ist daher, eine Summe aus dem Lokalisierungsfehler der korrekt zugeordneten Trajektorien und aus der Strafe für falsche und verfehlte Trajektorien zu bilden. Für die Gewichtung der Strafe wird ein zusätzlicher Parameter α eingeführt. Konkret wird die GOSPA-Metrik mit $c > 0$, $1 \leq p < \infty$ und $0 < \alpha \leq 2$ sowie den Teilmengen $X = \{x_1, \dots, x_{|X|}\}$ und $Y = \{y_1, \dots, y_{|Y|}\}$ für $|X| \leq |Y|$ definiert zu

$$d_p^{(c,\alpha)}(X, Y) := \left(\min_{\pi \in \Pi_{|Y|}} \sum_{i=1}^{|X|} d^{(c)}(x_i, y_{\pi(i)})^p + \frac{c^p}{\alpha} (|Y| - |X|) \right)^{1/p}. \quad (5-31)$$

Darin ist analog zur OSPA-Distanz die Distanz $d^{(c)}(x, y) := \min(c, d(x, y))$ mit der Metrik $d(x, y)$ für alle $x, y \in \mathbb{R}^N$. Wenn $|X| > |Y|$, dann gilt $d_p^{(c,\alpha)}(X, Y) = d_p^{(c,\alpha)}(Y, X)$. Die Parameter c und p haben dieselbe Bedeutung wie bei der OSPA- und OSPA-T-Metrik. Der Parameter c bestimmt die Cut-Off-Distanz und die Ordnung p gewichtet die Bestrafung bei Ausreißern.

Rahmathullah et al. legen für den Parameter $\alpha = 2$ fest, da dies wie eingangs erwähnt eine Aufteilung der GOSPA-Metrik in zwei Teile (Lokalisierungsfehler und Strafe für falsche und verfehlte Trajektorien) zulässt. Setze $\alpha = 2$, dann gilt für die umformulierte GOSPA-Metrik

$$d_p^{(c,2)}(X, Y) := \left[\min_{\gamma \in \Gamma} \sum_{(i,j) \in \gamma} d(x_i, y_j)^p + \frac{c^p}{2} (|X| + |Y| - 2|\gamma|) \right]^{1/p}, \quad (5-32)$$

wobei statt der Menge der Permutationen $\Pi_{|Y|}$ die Menge aller möglichen Zuordnungsmengen Γ verwendet wird. Eine Zuordnungsmenge γ zwischen den Teilmengen $\{1, \dots, |X|\}$ und $\{1, \dots, |Y|\}$ besitzt nach Rahmathullah et al. folgende Eigenschaften⁹⁶:

$$\gamma \subseteq \{1, \dots, |X|\} \times \{1, \dots, |Y|\}, (i, j), (i, j') \in \gamma \Rightarrow j = j' \text{ und } (i, j), (i', j) \in \gamma \Rightarrow i = i'.$$

5.4.6 Rahmathullah et al.-Metrik

In einer weiteren Veröffentlichung definieren Rahmathullah et al. die multidimensionale Zuordnungsdistanz $d_p^{(c,\gamma)}(\mathbf{X}, \mathbf{Y})$ mit $c > 0$, $1 \leq p < \infty$ und $\gamma > 0$ für alle $\mathbf{X}, \mathbf{Y} \in \mathcal{Y}$ wie folgt:⁹⁷

⁹⁵ Vgl. Rahmathullah, A. S. et al.: Generalized optimal sub-pattern assignment metric (2017).

⁹⁶ Rahmathullah, A. S. et al.: Generalized optimal sub-pattern assignment metric (2017), S. 2.

⁹⁷ Rahmathullah, A. S. et al.: Multidimensional assignment distance (2016), S. 6.

$$d_p^{(c,\gamma)}(\mathbf{X}, \mathbf{Y}) := \min_{\pi^k \in \Pi_{\mathbf{X}, \mathbf{Y}} \atop k=1, \dots, T} \left(\sum_{k=1}^T d_{\mathbf{X}, \mathbf{Y}}^k(\mathbf{X}, \mathbf{Y}, \pi^k) + \sum_{k=1}^{T-1} s_{\mathbf{X}, \mathbf{Y}}(\pi^k, \pi^{k+1}) \right)^{1/p}. \quad (5-33)$$

Diese Metrik ist in zwei Teile aufgeteilt. Die erste Summation addiert alle Lokalisierungsfehler $d_{\mathbf{X}, \mathbf{Y}}^k$, während die zweite Summation die Fehler $s_{\mathbf{X}, \mathbf{Y}}$ durch das Wechseln der Bezeichnung der Trajektorien berücksichtigt. Für eine Herleitung der Gleichungen sei auf Ramahtullah et al. verwiesen.⁹⁸ Der Parameter γ ist die Strafe für das Wechseln der Bezeichnung einer Trajektorie. Je größer γ ist, desto stärker ist die Gewichtung des Wechselns der Bezeichnung der Trajektorie.

5.4.7 OSPA-MT-Metrik

Vu und Evans entwickeln auf Basis der OSPA-Metrik die OSPA-Metrik for Multiple Tracks (OSPA-MT-Metrik). Diese beseitigt einige Schwachstellen der OSPA-T-Metrik von Ristic et al. Zu den Schwachstellen zählt nach Vu und Evans die Tatsache, dass die OSPA-T-Metrik nicht die Distanz zwischen zwei mit Trajektorien gefüllten endlichen Mengen misst, sondern die Distanz zwischen zwei mit Zuständen und jeweiligen Bezeichnungen gefüllten endlichen Mengen.^{99a} Ein Zustand mit einer Bezeichnung ist ein Vektor, der den Zustand des Ziels (z. B. Position, Geschwindigkeit, Ausdehnung des Objekts etc.) sowie dessen Bezeichnung („Label“) beinhaltet, während eine Trajektorie eine Sequenz von Zuständen eines Ziels ist. Die OSPA-MT-Distanz zwischen den Teilmengen ω und ω' ist mit $1 \leq p < \infty$ und $0 < \Delta < c$ definiert zu

$$d_{c,p}^\Delta(\omega, \omega') = \min\{d_{c,p}^\Delta(\overrightarrow{\omega'}, \overrightarrow{\omega}), d_{c,p}^\Delta(\overrightarrow{\omega}, \overrightarrow{\omega'})\} \quad (5-34)$$

mit der Quasi-OSPA-MT-Metrik

$$d_{c,p}^\Delta(\overrightarrow{\omega'}, \overrightarrow{\omega}) = \begin{cases} \min_{\lambda \in \mathcal{M}(L^{\omega'}, L_0^\omega)} \tilde{d}_{c,p}^{\Delta, \lambda}(\overrightarrow{\omega'}, \overrightarrow{\omega}), & \text{wenn } \omega, \omega' \neq \emptyset; \\ 0, & \text{wenn } \omega = \omega' = \emptyset; \\ c, & \text{sonst.} \end{cases} \quad (5-35)$$

Der Parameter Δ gewichtet den Fehler, wenn entweder mehrere Trajektorien einer Ground-Truth-Trajektorie zugewiesen werden oder umgekehrt. Vu und Evans beschreiben ein mehrstufiges Vorgehen, das die Berechnung der OSPA-MT-Metrik ermöglicht.^{99b}

5.4.8 Intersection-over-Union

Ganström et al.¹⁰⁰ stellen zur Leistungsbewertung von *Extended Object Trackern* (siehe Abschnitt 2.3.2) das Intersection-over-Union-Verfahren (IoU) vor. Da *Extended Object Tracker* neben der Position des Objekts weitere Zustände wie z. B. die räumliche Ausdehnung des Objekts schätzen, wird

⁹⁸ Vgl. Rahmathullah, A. S. et al.: Multidimensional assignment distance (2016), S. 5–6.

⁹⁹ Vgl. Vu, T. A.; Evans, R.: Optimal Subpattern Assignment Metric for Multiple Tracks (2018), a: S. 1; b: S. 7.

¹⁰⁰ Vgl. Granström, K. et al.: Extended Object Tracking (2016), S. 20.

eine Metrik benötigt, die einen Vergleich dieser Zustände mit den Ground-Truth-Zuständen ermöglicht. Die IoU-Metrik, auch Jaccard-Index genannt, ist definiert als Schnittmenge zweier Mengen dividiert durch die Vereinigungsmenge beider Mengen¹⁰¹, siehe Abbildung 5-3.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (5-36)$$

Die Berechnung des IoU-Wertes ist bspw. für die zweidimensionale Ausdehnung von geschätzten und Ground-Truth-Objekten unter Annahme einer rechteckigen Form möglich. Ganström et al. merken an, dass die Berechnung des IoU-Wertes aufwändig ist, wenn die Rechtecke nicht an den Koordinatenachsen ausgerichtet sind oder andere Formen betrachtet werden. Außerdem ist der IoU-Wert immer Null, falls sich die Mengen A und B nicht schneiden, unabhängig von der Distanz der Mengen A und B zueinander. Dies widerspricht der Intuition, dass sich der Fehler mit zunehmender Distanz der Mengen zueinander vergrößert.¹⁰⁰

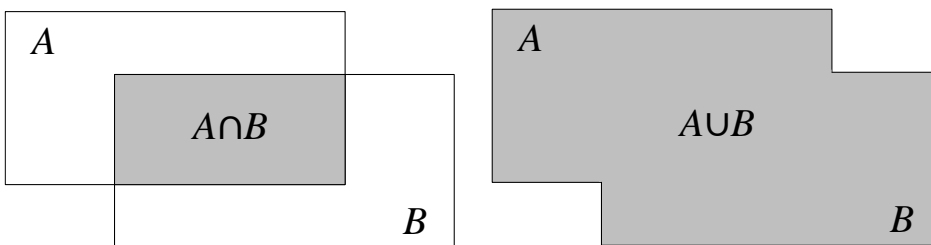


Abbildung 5-3: Schnittmenge und Vereinigungsmenge bei Objekten

5.4.9 Evaluation von Trackingalgorithmen nach Morton et al.

Morton et al.¹⁰² evaluieren die Leistungsfähigkeit unterschiedlicher Trackingalgorithmen mittels verschiedener Metriken. Ein Ansatz ist die Berechnung des *Root Mean Square Errors* (RMSE) zwischen der Ground-Truth- und der prädizierten Trajektorie. Wenn mehrere prädizierte Trajektorien mit einer Ground-Truth-Trajektorie korrespondieren, so wird eine Kombination aus allen geschätzten Trajektorien in der Berechnung berücksichtigt. Analog zur Berechnung des RMSE für die Position wird der RMSE der Geschwindigkeiten von Ground-Truth- und geschätztem Objekt berechnet. Diese Methode der Evaluierung erfordert ein manuelles Labeln der Trajektorien vor der Auswertung.

5.5 Klassifikation

Der allgemeine Ansatz zur Beurteilung der Leistungsfähigkeit eines Klassifizierungsalgorithmus ist die Berechnung einer Wahrheitsmatrix und mehreren daraus abgeleiteten Größen.^{103,104,105} Mittels der

¹⁰¹ Vgl. Jaccard, P.: *Distribution of the Flora* (1912), S. 39.

¹⁰² Vgl. Morton, P. et al.: *Evaluation of dynamic object tracking* (2011), S. 6–8.

¹⁰³ Vgl. Babahajiani, P. et al.: *Urban 3D segmentation* (2017), S. 689.

¹⁰⁴ Vgl. Dewan, A. et al.: *Classification for 3D LiDAR Data* (2017), S. 4.

¹⁰⁵ Vgl. Douillard, B. et al.: *Segmentation and Classification of 3D Point Clouds* (2014), S. 594–597.

Wahrheitsmatrix werden die Vorhersagen des Algorithmus mit den Referenzdaten verglichen und in einem anschließenden Berechnungsschritt unterschiedliche Werte wie Präzision, Trefferquote und Genauigkeit ermittelt. Wird eine einzelne Klasse k betrachtet, so ergeben sich nach Carrilho et al.¹⁰⁶ vier mögliche Fälle (siehe Tabelle 5-3):

- True Positive (TP): Die vorhergesagte Klasse des Algorithmus stimmt mit der Referenzklasse k überein.
- False Positive (FP): Der Algorithmus sagt die Klasse k vorher, obwohl das Referenzobjekt nicht der Klasse k angehört.
- True Negative (TN): Das vorhergesagte Objekt des Algorithmus und das Referenzobjekt werden beide nicht der Klasse k zugeordnet.
- False Negative (FN): Der Algorithmus ordnet dem Objekt die Klasse k nicht zu, obwohl das Referenzobjekt der Klasse k angehört.

Die Elemente der Wahrheitsmatrix beinhalten entweder die ganzzahlige Anzahl der Objekte oder die Häufigkeiten, wobei diese mit der Gesamtanzahl der Objekte der entsprechenden wahren Klasse normiert werden. Eine Normierung mit der Gesamtanzahl der Objekte aller Klassen ist auch möglich, jedoch unüblich.

Tabelle 5-3: Wahrheitsmatrix

		Wahre Kondition	
		Kondition positiv	Kondition negativ
Vorhergesagte Kondition	Vorhergesagte Kondition positiv	True Positive (TP)	False Positive (FP)
	Vorhergesagte Kondition negativ	False Negative (FN)	True Negative (TN)

Mit Kenntnis der TP , FP und FN wird die Präzision (*precision*) p

$$p = \frac{TP}{TP + FP} \quad (5-37)$$

und die Trefferquote (*recall*) r

$$r = \frac{TP}{TP + FN} \quad (5-38)$$

berechnet. Das harmonische Mittel der Präzision und der Trefferquote wird *F-Score* (oder *F₁-Score*, da gleiche Gewichtung von Präzision und Trefferquote) genannt

$$F\text{-Score} = \frac{2}{\frac{1}{p} + \frac{1}{r}} = \frac{2TP}{2TP + FP + FN} \quad (5-39)$$

¹⁰⁶ Vgl. Carrilho, A. C. et al.: Quality assessment for automatic LiDAR data classification methods (2017), S. 6773.

Diese Werte werden für alle betrachteten Klassen berechnet.

Eine andere Form der Wahrheitsmatrix stellen Aijazi et al.¹⁰⁷ vor. Die Autoren kombinieren die Wahrheitsmatrizen der einzelnen Klassen zu einer einzigen Wahrheitsmatrix, bei der die Anzahl der Objekte oder deren Häufigkeit der vorhergesagten Klassen den wahren Klassen zugeordnet sind (siehe beispielhafte Wahrheitsmatrix in Tabelle 5-4). Die Berechnung der vier möglichen Fälle für jede Klasse separat erfolgt zu (hier exemplarisch für die erste Zeile der Matrix S_{ij} mit $i = 1$ und der Gesamtanzahl der Klassen N):

$$TP = S_{11}, \quad (5-40)$$

$$FP = \sum_{j=2}^N S_{1j}, \quad (5-41)$$

$$TN = (1 - FP), \quad (5-42)$$

$$FN = (1 - TP). \quad (5-43)$$

Aijazi et al. berechnen daraus die individuelle Klassifizierungsgenauigkeit (*Classification Accuracy*) $CACC$ für jede der N Klassen

$$CACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (5-44)$$

und bilden aus allen $CACC$ den Mittelwert, die *Overall Classification Accuracy OCACC*

$$OCACC = \frac{1}{N} \sum_{i=1}^N CACC_i. \quad (5-45)$$

Dabei wird bei der individuellen und gesamten Klassifizierungsgenauigkeit ein Wert von Eins angestrebt. Wenn kein Objekt korrekt klassifiziert wird, nehmen die Klassifizierungsgenauigkeiten den Wert Null an.

Tabelle 5-4: Beispielhafte Wahrheitsmatrix mit Berechnung von $CACC$ und $OCACC$

		Wahre Klasse			$CACC$
		Pkw	Fußgänger	Lkw	
Vorhergesagte Klasse	Pkw	0,800	0,125	0,429	0,623
	Fußgänger	0,067	0,750	0	0,842
	Lkw	0,133	0,125	0,571	0,657
$OCACC$					0,707

¹⁰⁷ Vgl. Aijazi, A. et al.: Segmentation Based Classification (2013), S. 1634.

Eine andere Methode zur Berechnung der gesamten Klassifizierungsgenauigkeit wird von Douillard et al.¹⁰⁸ vorgeschlagen. Die Autoren ermitteln die Genauigkeit, indem sie alle *TP* der Wahrheitsmatrix (entspricht der Spur) summieren und durch die Gesamtanzahl aller Objekte dividieren. Für diese Berechnung wird die Wahrheitsmatrix mit den absoluten Anzahlen in den jeweiligen Klassen benötigt.

Für die Bewertung von binären Klassifikatoren werden *Receiver-Operating-Characteristics*-Kurven (ROC-Kurven) herangezogen. Bei diesen ist die TP-Rate über der FP-Rate aufgetragen. Die TP-Rate *tpr* ist definiert zu

$$tpr = \frac{TP}{Total\ Positives} = \frac{TP}{TP + FN} \quad (5-46)$$

und gleicht damit der Trefferquote (vgl. Gleichung (5-38)). Die FP-Rate *fpr* ist definiert zu

$$fpr = \frac{FP}{Total\ Negatives} = \frac{FP}{TN + FP} \quad (5-47)$$

repräsentiert also den Anteil der (wahren) negativen Klasse, der fälschlicherweise als positiv klassifiziert wird. Ein perfekter Klassifikator erzielt eine *tpr* = 1 bei *fpr* = 0 (Punkt A im ROC-Graphen in Abbildung 5-4), während ein schlechter Klassifikator im Worst Case eine *tpr* = 0 bei *fpr* = 1 aufweist (Punkt B im ROC-Graphen in Abbildung 5-4).¹⁰⁹ Punkt C im ROC-Graphen in Abbildung 5-4 stellt einen möglichen Zufalls-Klassifikator dar, wobei der Punkt abhängig von der Schätzung des Klassifikators auf der Diagonalen *tpr* = *fpr* wandert (hier 70 % positive Schätzung).^{110a} In Abbildung 5-4 sind weiterhin zwei beispielhafte ROC-Kurven möglicher Klassifikatoren eingetragen. Die schwarze (durchgezogene) Kurve ist dabei der roten (strichpunktierten) Kurve vorzuziehen, da sie bei der gleichen *fpr* eine höhere *tpr* aufweist bzw. bei der gleichen *tpr* weniger Falsch-Positiv Klassifikationen vornimmt. Der Vergleich unterschiedlicher ROC-Kurven erfolgt durch die Berechnung der Fläche unterhalb der ROC-Kurve (AUC, *Area under the ROC Curve*) im Intervall von null bis eins, je größer die Fläche, desto besser der Klassifikator. Die AUC liegt zwischen null und eins, wobei ein Wert größer als 0,5 in jedem Fall anzustreben ist, da dies das Ergebnis eines Zufalls-Klassifikators (Diagonale *tpr* = *fpr*) ist.^{110b}

Alle Überlegungen über die ROC-Kurven bis zu diesem Punkt gehen von einem binären Klassifikator aus, bei dem nur eine Klasse *n* existiert. Im Falle des Klassifizierungsalgorithmus für das Lidar-Sensormodell existieren mehrere Klassen, die der Algorithmus unterscheiden muss (z. B. Pkw, Lkw, Fußgänger, etc.). Fawcett stellt eine Methode vor, die eine Bewertung eines Multi-Klassen-Klassifikators ermöglicht.^{110c} Für jede Klasse $c_i \in C$ mit $i = 1, \dots, n$ wird eine ROC-Kurve berechnet. Die positive Klasse P_i ist c_i (Referenzklasse), während alle anderen Klassen zusammengefasst die negative Klasse N_i bilden

$$P_i = c_i, \quad (5-48)$$

¹⁰⁸ Vgl. Douillard, B. et al.: Segmentation and Classification of 3D Point Clouds (2014), S. 597.

¹⁰⁹ Vgl. Powers, D. M.: Receiver Operating Characteristics (2007), S. 4.

¹¹⁰ Vgl. Fawcett, T.: An introduction to ROC analysis (2006), a: S. 863; b: S. 868; c: S. 871-872.

$$N_i = \bigcup_{j \neq i} c_j \in C. \quad (5-49)$$

Ein Ansatz zur Berechnung einer kombinierten AUC stammt von Provost und Domingos.¹¹¹ Die Autoren definieren eine AUC, die den gewichteten Mittelwert aller AUC der einzelnen ROC-Kurven darstellt. Das Gewicht p einer Klasse ist abhängig von der Mächtigkeit der Klasse.

$$AUC_{\text{Total}} = \sum_{c_i \in C} AUC(c_i) \cdot p(c_i) \quad (5-50)$$

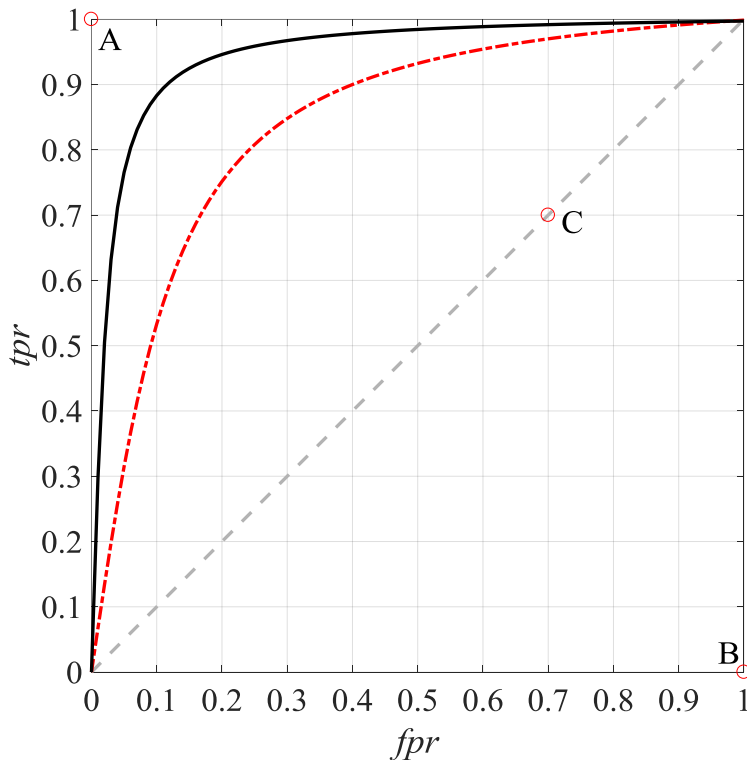


Abbildung 5-4: ROC-Graph

5.6 Gesamtmaß

Neben den zuvor beschriebenen Metriken für die einzelnen Verarbeitungsebenen wird überprüft, ob aus den einzelnen Metriken ein Gesamtmaß entwickelt werden kann. Außerdem wird die Anwendbarkeit dieses Gesamtmaßes untersucht. Dieses Gesamtmaß soll die Güte der Lidar-Sensorsimulation widerspiegeln. Die Idee hinter der Erstellung eines Gesamtmaßes ist es, dass die Ergebnisse mehrerer Metriken zu einer einzigen Zahl zusammengefasst werden. Somit kann durch Angabe dieser einzigen Zahl die Güte der Simulation abgeschätzt werden oder ein Vergleich unterschiedlicher Lidar-Sensorsimulationen vorgenommen werden.

¹¹¹ Vgl. Provost, F.; Domingos, P.: Improving Probability Estimation Trees (2000), S. 14.

Es existieren jedoch einige Gründe, die gegen eine Erstellung eines Gesamtmaßes sprechen. Diese werden im Folgenden aufgelistet. Zunächst ist durch das Angeben eines Gesamtmaßes nicht ersichtlich, bei welchen Verarbeitungsebenen gute oder schlechte Ergebnisse erzielt werden. Eine differenzierte Betrachtung der Metriken für die einzelnen Verarbeitungsebenen ist weiterhin notwendig. Ein weiterer Grund ist, dass nicht alle Metriken die gleiche Auswirkung auf das Gesamtergebnis der Simulation besitzen. Somit wäre eine Gewichtung der einzelnen Metriken bei der Erstellung des Gesamtmaßes notwendig. Diese Gewichtung wiederum ist nicht bekannt und es sind keine Erfahrungswerte mit dem Umgang und der Berechnung der Metriken vorhanden, die eine Gewichtung ermöglichen. Des Weiteren ist nicht bekannt, welche Werte einer Metrik eine gute oder schlechte Simulation auszeichnen (es sind nur Best-Case- oder Worst-Case-Werte bekannt), daher verfälschen möglicherweise die einzelnen Ergebnisse der Metriken das Gesamtmaß. Zuletzt ist der Wert eines Gesamtmaßes physikalisch schwierig interpretierbar, während der physikalische Hintergrund vieler Metriken eine eindeutige Interpretation zulässt (z. B. die Punktwolke-zu-Punktwolke-Distanz gibt den Abstand aller Punkte zueinander in Metern an).

Infolge der genannten Gründe wird kein Gesamtmaß entwickelt. Es wird jedoch eine Methode vorgeschlagen, die trotzdem die ganzheitliche Bewertung der Lidar-Sensorsimulation ermöglicht. Zunächst werden die Metriken für die einzelnen Verarbeitungsebenen berechnet. Anschließend wird untersucht, wie sich die Metriken der unterschiedlichen Verarbeitungsebenen verhalten. Hierfür wird vorgeschlagen, zunächst die Metriken der letzten Verarbeitungsebene zu betrachten. Falls bei diesen Metriken Auffälligkeiten wie z. B. besonders niedrige Werte vorhanden sind, werden die Metriken der vorherigen Verarbeitungsebene untersucht. Wenn die Metriken der vorherigen Verarbeitungsebene keine Auffälligkeiten vorweisen, dann liegt der Fehler eventuell in einer weiteren vorherigen Verarbeitungsebene. Wenn in allen vorherigen Verarbeitungsebenen keine außergewöhnlichen Ergebnisse gefunden werden, liegt der Fehler in der letzten Verarbeitungsebene. Auf diese Weise wird jede Verarbeitungsebene einzeln betrachtet und ähnlich der FMEA (siehe Kapitel 3) die einzelnen Verarbeitungsebenen bewertet. Anhand der Bewertung der einzelnen Verarbeitungsebenen wird die Güte der gesamten Simulation abgeschätzt.

Diese Vorgehensweise ermöglicht es außerdem, Fehler in einer Verarbeitungsebene zu entdecken, die sich nicht auf eine nachgelagerte Ebene oder den Ausgang der Lidar-Sensorsimulation auswirken. Bei der Betrachtung eines Gesamtmaßes wäre dies in diesem Fall schlechter und eine Zuordnung zur Ursache des schlechteren Ergebnisses fehlt.

5.7 Metrikenauswahl

In diesem Abschnitt werden die Überlegungen zur Metrikenauswahl erklärt. Für die Bewertung einer Lidar-Sensorsimulation liegen wenige Erfahrungswerte vor. Der Ansatz von Schärmann et al.¹¹² zur Validierung einer Lidar-Sensorsimulation ist die einzige Veröffentlichung, bei der sich die Autoren mit der gleichen Problematik der Bewertung einer Lidar-Sensorsimulation befassen. Daher werden

¹¹² Vgl. Schaermann, A. et al.: Validation of Vehicle Environment Sensor Models (2017).

die Metriken, die in dieser Veröffentlichung vorgestellt werden, in dieser Masterthesis ebenfalls berechnet und die Anwendbarkeit untersucht. Die weiteren in dieser Masterthesis vorgestellten Metriken sind nicht speziell für die Bewertung einer Lidar-Sensorsimulation entwickelt und daher liegen keine Erkenntnisse vor, ob sich diese für die Bewertung eignen oder ob eine Anwendung prinzipiell möglich ist. Daher wird wie folgt vorgegangen: Wenn möglich werden alle aufgelisteten Metriken berechnet, sofern ein passendes Szenario zur Verfügung steht. Anschließend werden die Ergebnisse diskutiert und eine Aussage über die Anwendbarkeit der Metriken getroffen. Falls eine Berechnung nicht möglich ist oder nicht durchgeführt wird, dann wird der Grund hierfür angegeben. Eine Aussage über die Eignung der Metriken zur Bewertung der Lidar-Sensorsimulation wird nicht getroffen, da die hierfür benötigten Anforderungen an die Simulation nicht vorhanden sind. Es werden jedoch Empfehlungen ausgesprochen, welche Metriken in weiteren Betrachtungen für die Bewertung einer Lidar-Sensorsimulation vermutlich geeignete Ergebnisse erzielen und welche Metriken für eine weitere Betrachtung nicht in Frage kommen, da beispielsweise eine Berechnung generell nicht möglich ist.

6 Exemplarische Berechnung der Metriken

In diesem Kapitel werden die Ergebnisse der vorgestellten Metriken an beispielhaften Szenarien gezeigt. Abschnitt 6.1 erläutert das Aufzeichnen der realen Messdaten. In diesem Abschnitt wird auch auf die Besonderheiten bei der Messung hinsichtlich der Abweichungen eingegangen. Abschnitt 6.2 befasst sich mit Erstellung der Szenarien in der Simulationsumgebung VTD, bevor in Abschnitt 6.3 die einzelnen Szenarien vorgestellt werden. Die Anwendung der Metriken für die einzelnen Verarbeitungsebenen sowie die kombinierte Betrachtung der Verarbeitungsebenen ist in Abschnitt 6.4 dokumentiert.

6.1 Aufzeichnen von realen Messdaten

Für die Bewertung der Lidar-Sensorsimulation und das Anwenden der Metriken werden Referenzdaten benötigt. Diese Messdaten werden in realen Fahrversuchen auf dem Testgelände der TU Darmstadt, dem August-Euler-Flugplatz in Griesheim, aufgezeichnet. Neben dem mit den Lidar-Sensoren ausgestatteten Ego-Fahrzeug Honda Accord kommen weitere Fahrzeuge zum Einsatz, die im Umfeld des Ego-Fahrzeugs platziert werden oder sich darin bewegen. Das Ego-Fahrzeug ist mit zwei der im Abschnitt 2.1.3 beschriebenen ibeo Lux 2010® Lidar-Sensoren ausgestattet, deren Daten mittels einer ECU fusioniert werden. Die Ausgabe der Daten erfolgt in Form einer idc-Datei, wobei die fusionierten Punktwolken relativ zum Referenzpunkt (Mitte der Hinterachse des Ego-Fahrzeugs) im kartesischen Koordinatensystem ausgegeben werden. Aufgrund der Ausgabe in kartesischen Koordinaten erfolgt die Berechnung aller Metriken ebenfalls im kartesischen Koordinatensystem, obwohl die Messdaten eines einzelnen Lidar-Sensors in Polarkoordinaten ausgegeben werden.

Das Ego-Fahrzeug ist außerdem mit dem hochpräzisen Kreismesssystem ADMA ausgestattet. Dies ermöglicht die Positionsbestimmung mittels DGPS. Ein zweiter ADMA ist in einem weiteren Fahrzeug (BMW i3) verbaut. Die Kommunikation der beiden ADMA erfolgt über ein Wi-Fi-Kit, das die Daten vom Delta-Fahrzeug an das Ego-Fahrzeug via WLAN sendet und im Ego-Fahrzeug speichert. Zu den übermittelten Daten zählt unter anderem die Relativposition und -orientierung der beiden Points-of-Interest (jeweils Mitte der Hinterachse der Fahrzeuge) zueinander. Die Relativentfernung des Delta-Fahrzeuges zum Ego-Fahrzeug wird in eine X- und Y-Komponente aufgeteilt.

Abbildung 6-1 zeigt die Anordnung der mit den ADMA ausgestatteten Fahrzeugen mit den zugehörigen Distanzen und der Orientierung. Die Vorzeichenkonvention folgt der DIN ISO 8855, in der die X-Achse in Richtung der Fahrzeuglängsachse in Fahrtrichtung nach vorne als positiv definiert ist, die Y-Achse senkrecht zur Fahrzeuglängsmittlebene und in Fahrtrichtung nach links gerichtet ist und die Z-Achse nach oben zeigt.¹¹³ Die Positionen der weiteren Fahrzeuge, die nicht mit einem ADMA ausgerüstet sind, werden mit einem Maßband relativ zu den Fahrzeugen mit einem ADMA vermessen.

¹¹³ Vgl. DIN Deutsches Institut für Normung e. V.: DIN ISO 8855 (2013), S. 6.

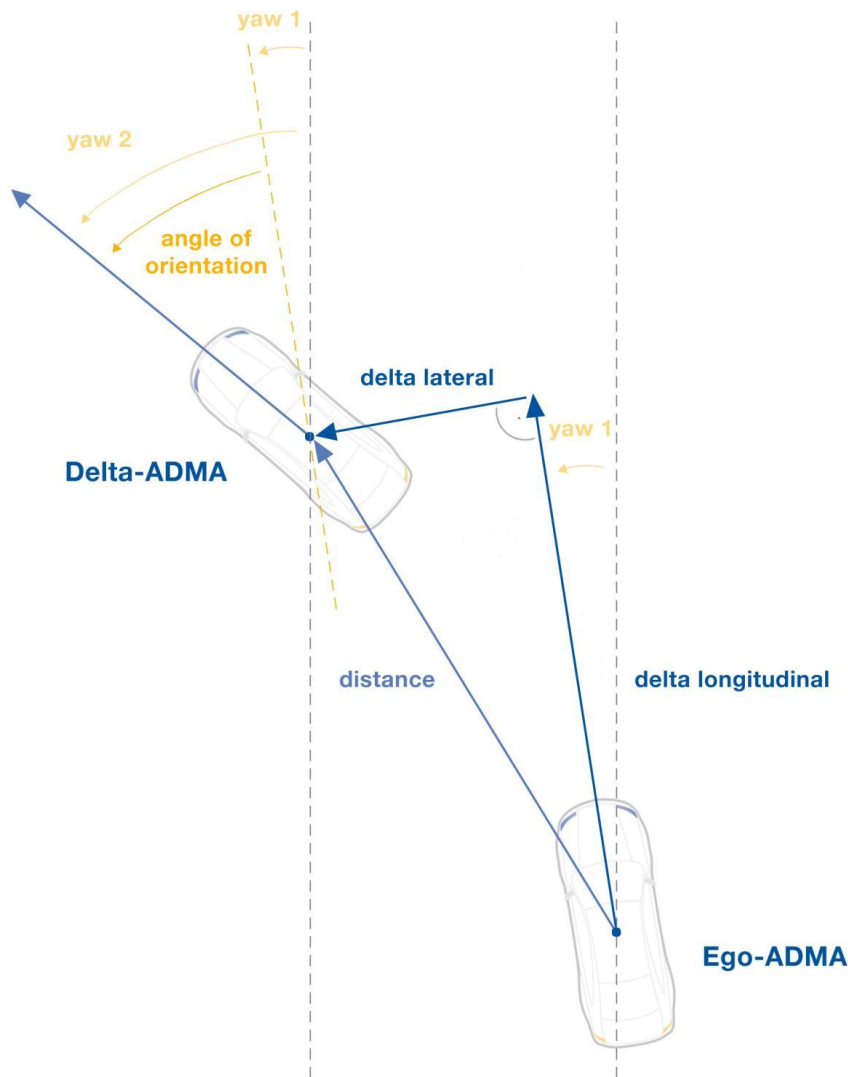


Abbildung 6-1: Relativposition und -orientierung des ADMA¹¹⁴

6.1.1 Ermittlung der Winkelabweichung der Lidar-Messdaten

Bei der Validierung der Messdaten der Lidar-Sensoren ist aufgefallen, dass die gemessenen Punktwolken einen horizontalen Winkelversatz gegenüber den vermuteten Ground-Truth-Positionen der Objekte im Sichtbereich der Sensoren aufweisen. Mit dem folgenden Versuch wird die Winkelabweichung der Lidar-Sensoren quantifiziert, eine schematische Darstellung des Versuchsaufbaus befindet sich in Abbildung 6-2. Es wird ein *Corner Reflector* in bestimmten Abständen zum Versuchsfahrzeug positioniert, dessen Lage zum Versuchsfahrzeug mit einem Maßband vermessen wird (dient als Ground Truth). Der *Corner Reflector* besteht aus drei senkrecht aufeinander stehenden Metallplatten und reflektiert jeden einfallenden Lichtstrahl dreimal, sodass dieser eine Richtungsumkehr vollzieht und in die Richtung des Lidar-Sensors zurückgeworfen wird. Somit wird sichergestellt, dass

¹¹⁴ Mit Änderungen entnommen aus GeneSys Elektronik GmbH: ADMA 3.0 Manual (2018), S. 71.

trotz der geringen Größe des *Corner Reflectors* dieser auch in größerer Entfernung vom Sensor detektiert wird. Die geringe Größe des *Corner Reflectors* ist notwendig, da dieser sonst von zu vielen Strahlen getroffen wird und die Position des *Corner Reflectors* nur ungenau geschätzt wird.

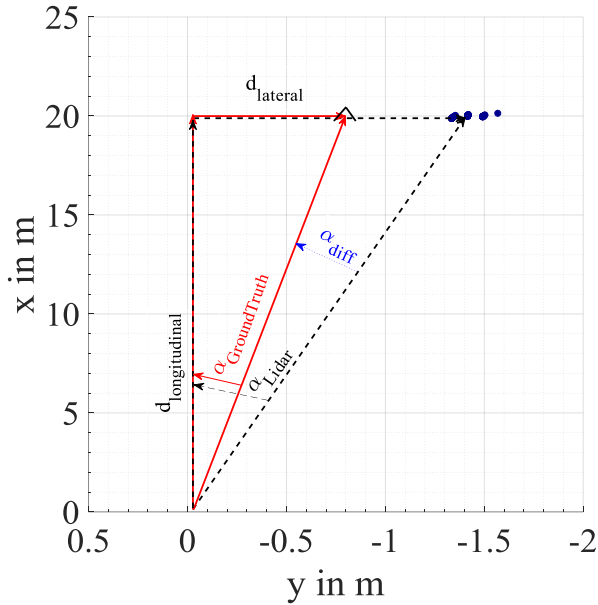


Abbildung 6-2: Vergleich Position des Corner Reflectors, Ground-Truth-Position (^) und Lidar-Punktwolke (•)

Der *Corner Reflector* wird in circa 1 m Höhe in verschiedenen longitudinalen Abständen vor dem Versuchsfahrzeug platziert. Der laterale Abstand zur Fahrzeuglängsmittlebene beträgt bei allen Versuchen -0,8 m (Ausnahme: -0,92 m bei 50 m longitudinalem Abstand), da der *Corner Reflector* und das Versuchsfahrzeug an einer Fahrstreifenmarkierung ausgerichtet sind. Der Winkel zwischen der Verbindungslinie des Fahrzeugs zum *Corner Reflector* und der Fahrzeuglängsachse beträgt demnach

$$\alpha = \tan^{-1} \frac{d_{\text{lateral}}}{d_{\text{longitudinal}}}. \quad (6-1)$$

Dieser Winkel wird ebenfalls für die gemessene Lidar-Punktwolke des *Corner Reflectors* bestimmt. Hierzu werden aus der gesamten Punktwolke die Punkte extrahiert, die eindeutig dem *Corner Reflector* zugeordnet werden. Die Entfernungen d_{lateral} und $d_{\text{longitudinal}}$ folgen aus der Berechnung des Schwerpunkts der extrahierten Punktwolke. Aus den Winkelpaaren α_{Lidar} und $\alpha_{\text{Ground Truth}}$ wird für jede Distanz die Differenz gebildet, wobei der Winkel α_{Lidar} aus drei konsekutiven Messungen gemittelt wird. Diese Winkeldifferenz gibt die Abweichung an, die die Punktwolke der Lidar-Sensoren gegenüber der Ground Truth aufweist (siehe Tabelle 6-1 für alle Einzelwerte).

Tabelle 6-1: Vergleich gemessene und tatsächliche Position des Corner Reflectors

Ground Truth			Gemessene Lidar-Punktwolke			Differenz
$d_{\text{longitudinal}}$ [m]	d_{lateral} [m]	$\alpha_{\text{Ground Truth}}$ [°]	$d_{\text{longitudinal}}$ [m]	d_{lateral} [m]	α_{Lidar} [°]	α_{diff} [°]
20	-0,8	-2,2906	19,9696	-1,4277	-4,0893	1,7987
30	-0,8	-1,5275	30,0262	-1,6759	-3,1946	1,6671
50	-0,96	-1,0541	50,1051	-2,4157	-2,7603	1,7061

Da diese Abweichung nicht bei allen Distanzen konstant ist, wird die Winkeldifferenz der drei Distanzen gemittelt. Die resultierende gemittelte Winkeldifferenz beträgt $\alpha_{\text{diff}} = 1,7204^\circ$. Alle mit den realen Lidar-Sensoren generierten Punktwolken werden vor der Anwendung der Metriken durch eine Koordinatentransformation mit der Rotationsmatrix

$$R_z(\alpha_{\text{diff}}) = \begin{pmatrix} \cos \alpha_{\text{diff}} & \sin \alpha_{\text{diff}} & 0 \\ -\sin \alpha_{\text{diff}} & \cos \alpha_{\text{diff}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6-2)$$

um die Z-Achse mit dem zuvor bestimmten Winkel α_{diff} rotiert.

6.1.2 Ermittlung der Abweichung der Delta-Entfernung

Bei der Erstellung der Szenarien in der Simulation wird für die genaue Bestimmung der relativen Positionen der Fahrzeuge zueinander auf die Delta-Daten des ADMA zurückgegriffen. In einer vorherigen Masterthesis bei FZD von Ngo werden die Delta-Daten ebenfalls verwendet, bei der Validierung der Messdaten hat sich jedoch ein Großteil der Delta-Daten als fehlerbehaftet herausgestellt. Da eine Anwendung der Metriken zur Bewertung der Lidar-Sensorsimulation nur sinnvoll ist, wenn die realen Szenarien in der Simulation mit größtmöglicher Genauigkeit nachgestellt werden, werden die Delta-Daten überprüft. Hierzu werden vor der Messdatenaufnahme mehrere Versuche durchgeführt mit dem Ziel, die Delta-Daten zu validieren.

In einem ersten Versuch wird die Distanz zwischen dem Versuchsfahrzeug und dem mit dem Delta-ADMA ausgestatteten Fahrzeug zusätzlich mit einem Laser-Entfernungsmesser ermittelt. Die Delta-Distanz, die von der Ego-ADMA ausgegeben wird, entspricht dabei der Distanz zwischen den Mittelpunkt der Hinterachsen der Fahrzeuge. Mit dem Laser-Entfernungsmesser wird die Distanz vom Radar des Versuchsfahrzeugs zum Heck des anderen Fahrzeugs (BMW i3) gemessen. Diese Distanz wird korrigiert um den Abstand vom Heck des i3 zu dessen Hinterachse (735 mm) sowie um den Abstand vom Radar des Versuchsfahrzeugs zu dessen Hinterachse (3600 mm), also insgesamt 4335 mm. Eine Zusammenfassung der Ergebnisse befindet sich in Tabelle 6-2. Die Differenz zwischen den Distanzen beträgt 18 bzw. 25 mm und liegt innerhalb der Messgenauigkeit des ADMA mit DGPS und des Laser-Entfernungsmessers.

Tabelle 6-2: Überprüfung der ADMA-Distanz, alle Werte in mm

Laser-Entfernungsmesser			ADMA	Differenz
Messwert	Korrekturwert	Korrigierter Messwert	Delta-Distanz (Delta_X)	
6073	4335	10408	10390	18
20050	4335	24385	24360	25

In einem weiteren Versuch wird die Delta-Distanz mit der Distanz, die sich aus der Berechnung der Differenz zweier GPS-Punkte ergibt, verglichen. Somit wird sichergestellt, dass die interne Berechnung der Delta-Distanz mit den einzelnen Messdaten in den Versuchsfahrzeugen übereinstimmt. Die Berechnung der kürzesten Distanz zwischen zwei GPS-Punkten, von denen jeweils der Breitengrad

ϕ_1 bzw. ϕ_2 und Längengrad λ_1 bzw. λ_2 bekannt sind, erfolgt unter Anwendung der Haversine-Funktion

$$\text{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right). \quad (6-3)$$

Die kürzeste Distanz d zwischen zwei GPS-Punkten auf einer sphärischen Kugel befindet sich auf einer Orthodrome und berechnet sich zu

$$d = r\alpha. \quad (6-4)$$

Für die Berechnung des zentralen Winkels zwischen den GPS-Punkten gilt

$$\alpha = 2 \operatorname{atan2}\left(\sqrt{a}, \sqrt{(1-a)}\right) \quad (6-5)$$

mit

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cos\phi_2 \sin^2\left(\frac{\Delta\lambda}{2}\right). \quad (6-6)$$

Der Erdradius in Gleichung (6-4) wird unter Annahme eines sphäroidischen Erdmodells mit der großen Halbachse $r_1 = 6378,1370$ km und der kleinen Halbachse $r_2 = 6356,7523$ km in Abhängigkeit des Breitengrades angenähert zu

$$r(\phi) = \sqrt{\frac{(r_1^2 \cos^2 \phi) + (r_2^2 \sin^2 \phi)}{(r_1 \cos \phi)^2 + (r_2 \sin \phi)^2}}. \quad (6-7)$$

Für die Überprüfung der Delta-Distanz wird der Honda Accord (Ego-Fahrzeug) an der Start- und Landebahn ausgerichtet, während der BMW i3 sich in Längsrichtung langsam von dem statischen Ego-Fahrzeug entfernt und sich anschließend wieder nähert (siehe Abbildung 6-3 Diagramm oben links). Die aufgezeichneten Delta-Daten (rote Kreise) werden mit der Differenz der GPS-Punkte (schwarze Punkte), die mit den vorgestellten Gleichungen berechnet wird, verglichen. Da die Differenzen im Verhältnis zur relativen Position nicht groß sind, liegen die jeweiligen Punkte fast übereinander. Um dies besser zu veranschaulichen, wird von den durch den ADMA und den manuell berechneten relativen Positionen die Differenz gebildet (siehe Abbildung 6-3 Diagramm unten links). Durch Kombination der beiden Diagramme entsteht ein drittes Diagramm, welches die Differenz der relativen Position in Abhängigkeit von der relativen Position zeigt (siehe Abbildung 6-3 Diagramm rechts). Die Abweichung zwischen den Berechnungen nimmt annähernd linear mit der relativen Position zu, außerdem ist eine geringe Hysterese zu erkennen. Ein Grund für die Abweichung zwischen der Berechnung der Delta-Distanz durch den ADMA und der manuellen Berechnung der Delta-Distanz ist möglicherweise die unterschiedliche Bestimmung des lokalen Erdradius. Die Abweichung in den für die Szenarien relevanten Entfernungen (meistens < 50 m) ist mit Differenz-Werten $\leq 0,2$ m jedoch akzeptabel. Nach Rücksprache mit dem Support des Herstellers des ADMA erfolgt die interne Berechnung der Delta-Distanzen nicht mit der Haversine-Funktion, da sich diese Berechnung eher

für größere Distanzen eignet und bei geringen Distanzen zu ungenau ist.¹¹⁵ Dies erklärt die Abweichung in Abbildung 6-3.

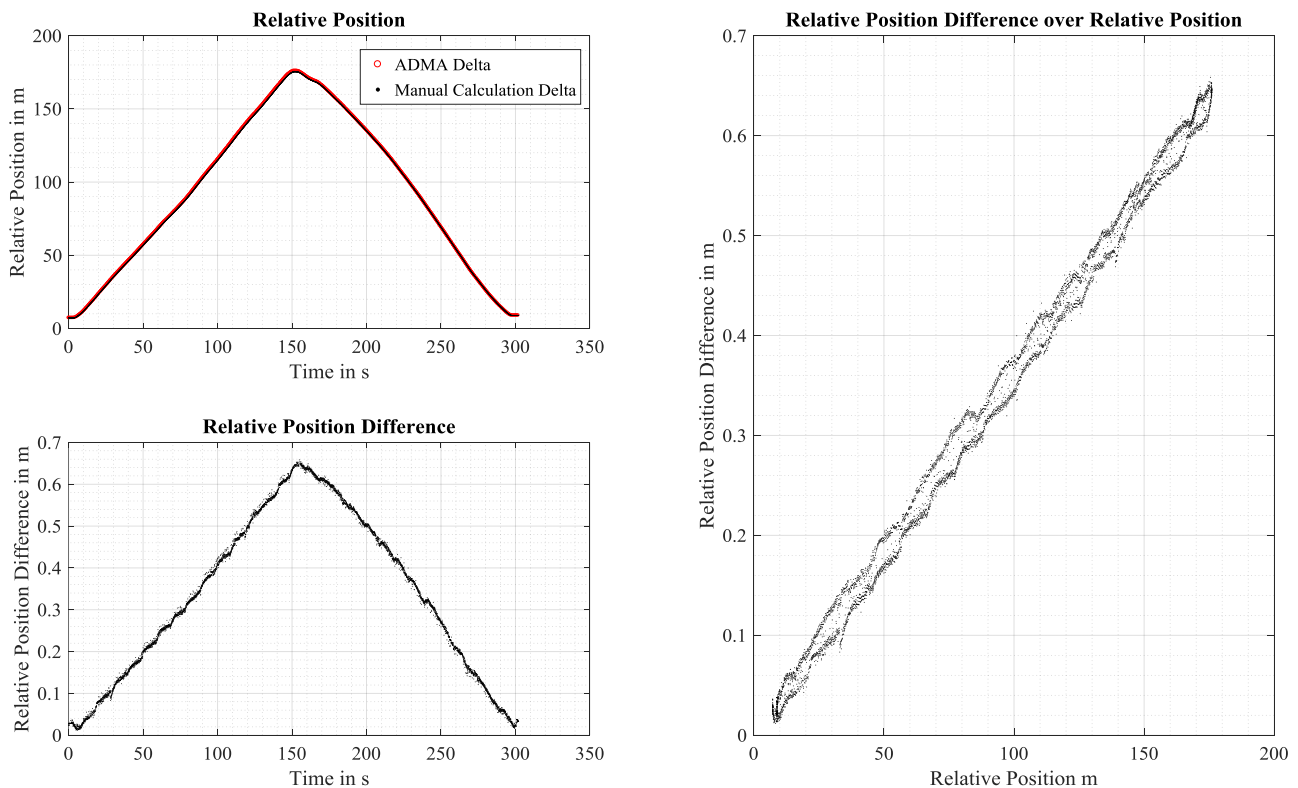


Abbildung 6-3: Vergleich ADMA-Delta-Distanzen mit manuell berechneten Delta-Distanzen, dynamischer Versuch

Um einen möglichen Fehler durch den sich bewegenden BMW i3 auszuschließen, wird der gleiche Versuch mit statischen Fahrzeugen und verschiedenen Abständen zueinander wiederholt. In Tabelle 6-3 sind die ADMA-Delta-Distanzen sowie die manuell berechneten Delta-Distanzen eingetragen. Die Differenzen im statischen Versuch sind geringer als die Differenzen des dynamischen Versuchs.

Mit den statischen und dynamischen Versuchen zur Überprüfung der Delta-Distanz sowie der Überprüfung mittels Laser-Entfernungsmesser wird gezeigt, dass die Genauigkeit der intern ermittelten ADMA-Delta-Distanzen für die Bewertung der Lidar-Sensorsimulation ausreicht. Daher wird bei der Erstellung der Szenarien in VTD auf die Delta-Distanzen zurückgegriffen, sofern diese vorhanden sind. Wichtig ist jedoch bei jeder Messung, dass der ADMA vor dem Aufzeichnen der Daten kalibriert wird, bspw. durch das Abfahren von Kreisen in Form einer Acht.¹¹⁶ Falls der ADMA kurzzeitig abgeschaltet ist, z. B. aufgrund einer Spannungsunterbrechung des Fahrzeuges, ist es zwingend notwendig, den ADMA neu zu kalibrieren.

¹¹⁵ Vgl. Stäuber, P.: Radiusberechnung nach WGS84 (2018).

¹¹⁶ Vgl. GeneSys Elektronik GmbH: ADMA 3.0 Manual (2018), S. 52.

Tabelle 6-3: Vergleich der ADMA-Delta-Distanzen mit manuell berechneten Delta-Distanzen, statischer Versuch

ADMA-Delta-Distanzen	Manuell berechnete Delta-Distanzen	Differenzen
09,970 m	10,076 m	-0,106 m
19,993 m	20,080 m	-0,087 m
29,975 m	30,049 m	-0,074 m
39,956 m	40,004 m	-0,048 m
50,017 m	50,055 m	-0,038 m

6.2 Erstellen der Szenarien in VTD

Die in den realen Fahrversuchen aufgezeichneten Szenarien werden in das Simulationstool VTD übertragen. Mithilfe der Daten der beiden ADMA sowie den Messungen mittels Maßband werden die Szenarien im Szenario-Editor nachgestellt. Hierbei ist auf mehrere Einzelheiten zu achten. Zunächst erfolgt die Auswahl der Fahrzeuge. Da nicht alle Fahrzeuge, die in den realen Fahrversuchen verwendet werden, in VTD verfügbar sind, wird auf geometrisch ähnliche Fahrzeuge zurückgegriffen. In der nachfolgenden Tabelle 6-4 sind die Fahrzeuge aufgelistet, die in den realen Fahrversuchen verwendet werden, sowie deren jeweiliges Äquivalent in VTD. Das Ego-Fahrzeug (Honda Accord) ist nicht enthalten, da sich dieses nicht im Sichtbereich der Lidar-Sensoren befindet. Wichtig ist beim Ego-Fahrzeug, dass die Position und Ausrichtung der Lidar-Sensoren mit der realen Einbauposition und -orientierung übereinstimmt (vgl. Tabelle 2-1). Außerdem sind das Fahrdynamik-Modell sowie die Fahrbahn-Topologie in der Simulation von Bedeutung.

Tabelle 6-4: Fahrzeugäquivalente in VTD

Fahrzeug im realen Fahrversuch	Fahrzeugäquivalent in VTD
BMW i3	BMW i3
BMW 535i	Audi A6
BMW M3	Audi S5
MB Vito	VW Touran
Opel Corsa	Audi A3
VW Caddy	VW Touran
VW Crafter	MB Sprinter
VW Passat	VW Passat
VW Golf Cabrio	VW Golf

Nach der Auswahl der Fahrzeuge werden diese gemäß den ADMA-Daten auf dem virtuellen Abbild des August-Euler-Flugplatzes platziert. Bei den statischen Szenarien wird das Ego-Fahrzeug längs auf einem Fahrstreifen ausgerichtet und, falls vorhanden, um die laterale Delta-Distanz versetzt. Der BMW i3 wird auf dem gleichen Fahrstreifen platziert, jedoch um die longitudinale Delta-Distanz versetzt und um den relativen Orientierungswinkel gedreht. Wenn weitere Fahrzeuge neben dem

BMW i3 vorhanden sind, werden diese analog zum BMW i3 mit dem per Maßband gemessenen lateralen Offset platziert.

Für die Erstellung der dynamischen Szenarien zur Bewertung des Trackingalgorithmus werden die gsda-Dateien des ADMA in xosc-Dateien umgewandelt, damit diese in VTD eingelesen werden können. Außerdem erfolgt die Transformation der GPS-Koordinaten in das relative Koordinatensystem von VTD mithilfe der Gleichungen (6-4) bis (6-6).

6.3 Beschreibung der Szenarien

Für die Bewertung der Punktwolken und die daraus generierten Occupancy Grids werden statische Szenarien benötigt. Es werden insgesamt drei verschiedene statische Szenarien in realen Fahrversuchen aufgezeichnet und in der Simulation nachgestellt. Die Distanz vom Ego-Fahrzeug zu den Fahrzeugen im Sichtbereich der Lidar-Sensoren wird dabei in den unterschiedlichen Szenarien variiert, um den Einfluss des Abstands auf die Qualität der Simulation zu untersuchen.

Bei dem ersten statischen Szenario (Name: „i3 statisch Heck/Front/Seite“) wird der BMW i3 in verschiedenen longitudinalen Abständen zwischen 10 und 50 m zum Ego-Fahrzeug vor diesem platziert. Der Versuch wird zudem noch variiert, indem der BMW i3 entweder mit dem Heck, der Front oder der Seite zum Ego-Fahrzeug ausgerichtet ist. Bei den Versuchen mit Längsausrichtung des BMW i3 existiert kein lateraler Versatz von Ego-Fahrzeug und BMW i3. Bei Querausrichtung des BMW i3 liegt dessen Vorderachse auf der Verlängerung der Geraden, die durch die rechte Seite des Ego-Fahrzeugs gebildet wird. Eine schematische Darstellung des Versuchsaufbaus mit den jeweiligen Distanzen ist in Abbildung 6-4 gezeigt. Darin werden die Distanzen zwischen den Vorderachsen (VA) oder Hinterachsen (HA) oder den Seiten der Fahrzeuge gemessen. Die Distanzen zwischen BMW i3 und Ego-Fahrzeug werden bei diesem Versuch mit einem Maßband ermittelt, da zum Zeitpunkt des Versuchs kein ADMA im BMW i3 verbaut ist. Aufgrund des fehlenden Delta-ADMA sind keine Informationen über den relativen Orientierungswinkel und den lateralen Versatz der Fahrzeuge bekannt. Die Fahrzeuge werden visuell so ausgerichtet, dass die beiden Werte so gering wie möglich werden.

Das zweite statische Szenario (Name: „Stauende 2 Fahrzeuge“) beinhaltet neben dem BMW i3 noch den VW Crafter. Beide Fahrzeuge werden in longitudinalen Abständen zwischen 10 und 50 m zum Ego-Fahrzeug vor diesem nebeneinander platziert. Da in diesem Versuch der Delta-ADMA im BMW i3 verbaut ist, liegen Informationen über die Relativposition und -orientierung von BMW i3 und Ego-Fahrzeug vor. Der VW Crafter wird visuell neben dem BMW i3 ausgerichtet, sodass die Hinterachsen der Fahrzeuge auf einer Linie liegen. Der Abstand zwischen den Außenkanten der Räder des BMW i3 und VW Crafter wird mittels Maßband gemessen. Es wird angenommen, dass beide Fahrzeuge (BMW i3 und VW Crafter) parallel zueinander stehen und somit die gleiche Relativorientierung zum Ego-Fahrzeug aufweisen. Eine Übersicht über den schematischen Aufbau dieses Szenarios mit den zugehörigen Distanzen und Winkeln befindet sich in Abbildung 6-5. Dieses Szenario wird in ähnlicher Weise mit drei statischen Fahrzeugen durchgeführt. Die aufgezeichneten Daten sind jedoch nicht verwendbar, da der Delta-ADMA im BMW i3 bei diesem Versuch nicht korrekt kalibriert ist.

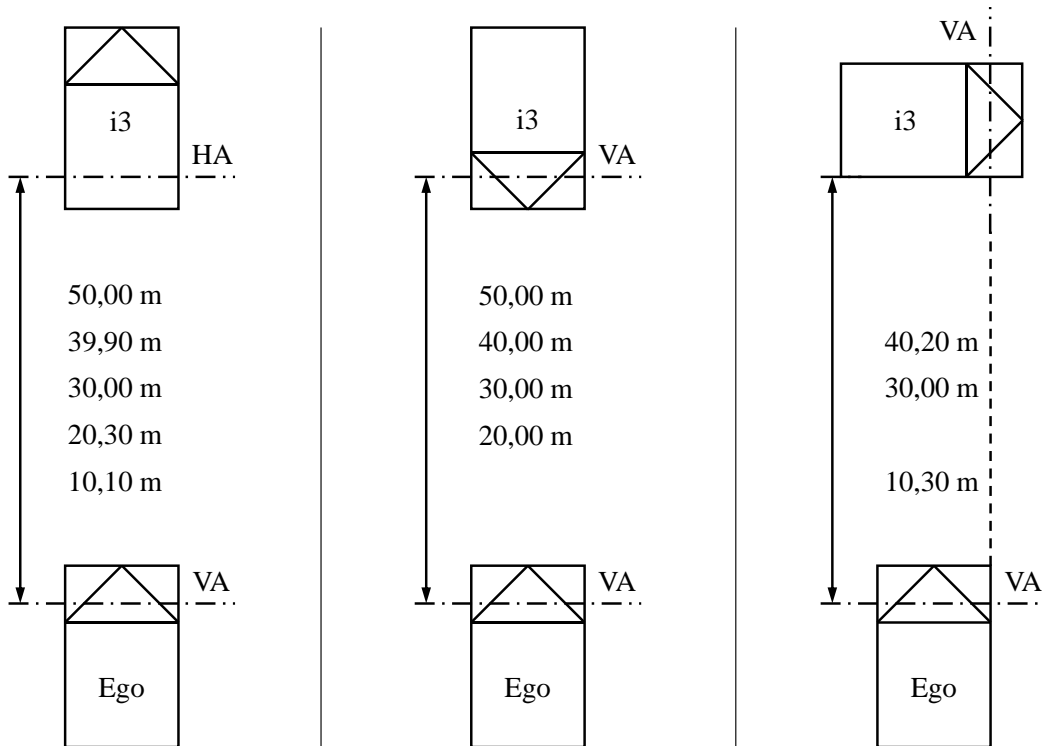


Abbildung 6-4: Aufbau des Szenarios BMW „i3 statisch Heck/Front/Seite“ mit Abständen

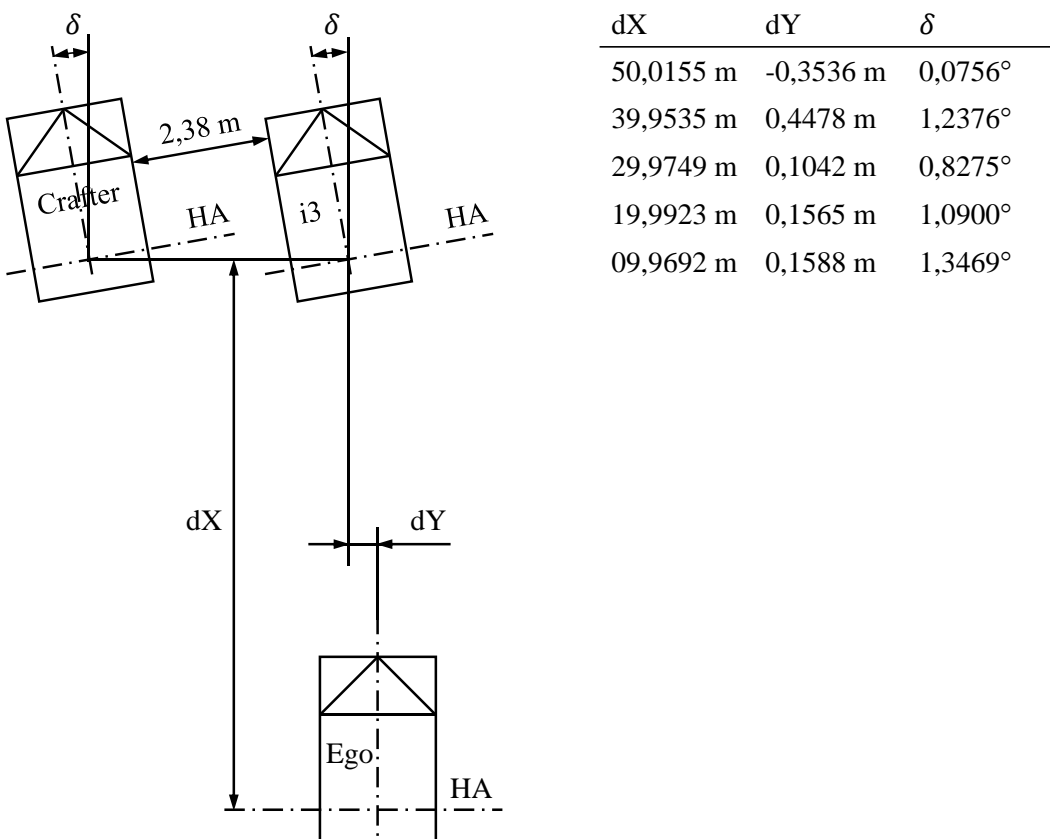


Abbildung 6-5: Aufbau des Szenarios „Stauende 2 Fahrzeuge“ mit Abständen und Orientierung

In einem dritten statischen Szenario (Name: „parkende Autos“) wird überprüft, wie die Qualität der Simulation bei Objekten mit großen Abständen zum Sensor ist. Hierzu werden sieben Fahrzeuge hintereinander mit einem lateralen Versatz zum Ego-Fahrzeug am Rand der Start- und Landebahn aufgereiht. Zudem wird in großer Entfernung zum Ego-Fahrzeug vor diesem ein achttes Fahrzeug platziert. Die relativen Distanzen zwischen den Fahrzeugen werden mit steigendem Abstand vom Ego-Fahrzeug größer, damit alle Fahrzeuge mit der gesamten Fahrzeugbreite im Sichtbereich der Lidar-Sensoren sind. Abbildung 6-6 zeigt eine nicht maßstabsgetreue Darstellung des Szenarios. Alle Distanzen sind in mm angegeben. Die sieben Fahrzeuge in Reihe sind dabei alle mit der Außenkante des rechten Hinterrades an einer Linie ausgerichtet. Da einige in diesem Szenario verwendete Fahrzeuge nicht in VTD vorhanden sind, werden geometrisch ähnliche Fahrzeuge in der Simulation verwendet, siehe hierzu Tabelle 6-4.

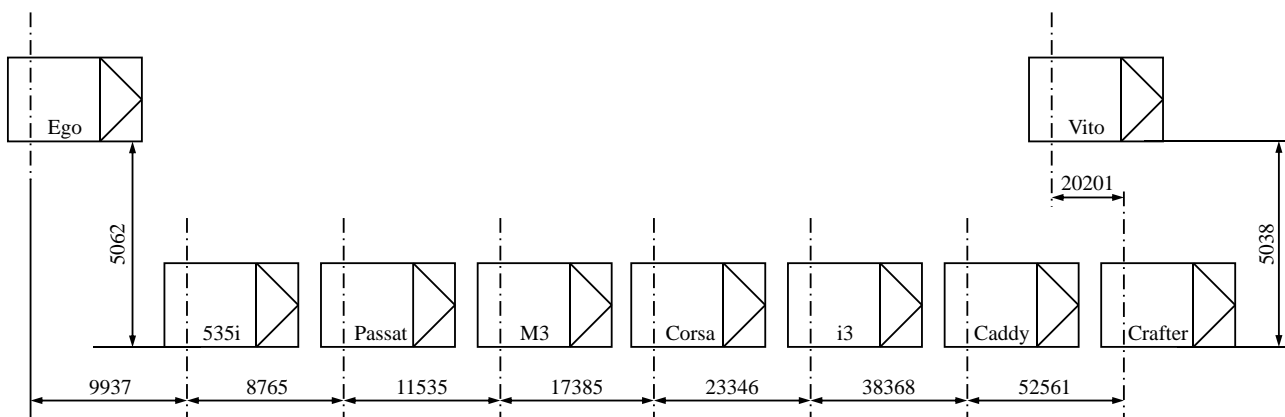


Abbildung 6-6: Aufbau des Szenarios „parkende Autos“, alle Maße in mm

Neben den statischen Szenarien zur Bewertung der Punktwolken werden dynamische Versuchsfahrten durchgeführt, die eine Bewertung des Trackingalgorithmus ermöglichen. Es werden zwei unterschiedliche Szenarien betrachtet. Der erste Versuch (Bezeichnung: „Stauende dynamisch“) wurde in einer vorherigen Masterthesis aufgezeichnet und in dieser Masterthesis neu simuliert. Bei diesem Versuch stehen drei Fahrzeuge (VW Golf, VW Caddy und BMW i3) nebeneinander und das Ego-Fahrzeug nähert sich diesen mit einer Geschwindigkeit von 100 km/h von hinten. Das Ego-Fahrzeug bremst so ab, dass es kurz hinter den drei Fahrzeugen zum Stehen kommt (siehe Abbildung 6-7). Die Startposition des Ego-Fahrzeuges wird so gewählt, dass die Distanz zwischen dem Ego-Fahrzeug und den drei Fahrzeugen größer ist als die maximale Sensorreichweite. Mit diesem Versuch wird die Trennfähigkeit des Trackingalgorithmus überprüft. Des Weiteren erfolgt die kombinierte Betrachtung der Verarbeitungsebenen mit diesem Versuch. Beim zweiten Szenario (Bezeichnung: „i3 Schrittgeschwindigkeit“) ist das Ego-Fahrzeug dauerhaft im Stillstand, während sich der BMW i3 dem Ego-Fahrzeug von vorne aus circa 500 Metern mit Schrittgeschwindigkeit nähert. Der Versuch wird beendet, sobald sich der i3 wenige Meter vor dem Ego-Fahrzeug befindet. Mit diesem Szenario wird überprüft, wie sich der Trackingalgorithmus verhält, wenn die Bewegungsgeschwindigkeit des getrackten Fahrzeuges gering ist.

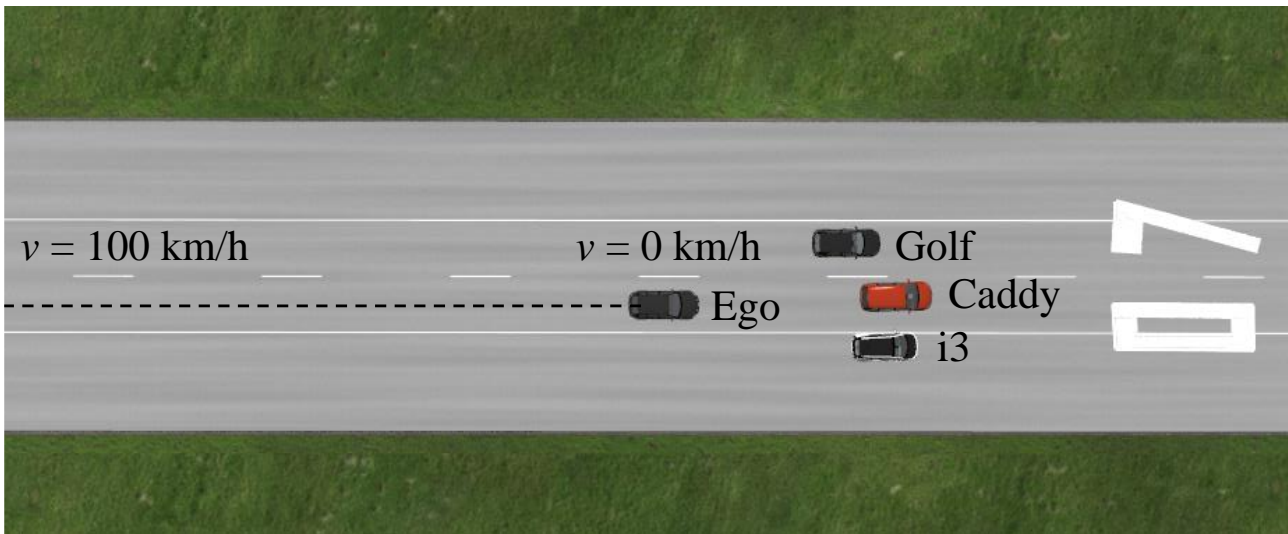


Abbildung 6-7: Szenario „Stauende dynamisch“, Screenshot aus VTD

6.4 Anwendung der Metriken

Nach dem Aufzeichnen der Szenarien in den realen Fahrversuchen und dem anschließenden Simulieren der gleichen Szenarien in VTD wird das Simulationsergebnis mittels ausgewählter Metriken bewertet. Neben den in Kapitel 5 vorgestellten Metriken werden weitere abgewandelte Metriken berechnet, um die Qualität der Simulation zu bewerten. Ziel der Berechnung der Metriken ist es, zum einen eine Aussage über die Qualität der Simulation und zum anderen eine Aussage über die Anwendbarkeit der einzelnen Metriken für die Bewertung einer Sensorsimulation zu treffen.

In diesem Abschnitt werden exemplarisch nur die Verarbeitungsebenen Punktwolken und Tracking betrachtet. Der Grund dafür ist, dass die Verarbeitungsebene Klassifikation zum Zeitpunkt des Verfassens dieser Masterthesis noch nicht für eine Bewertung verwendbar ist. Der Klassifizierungsalgorithmus ist nur in der Lage, zwischen den Klassen Fahrzeug und Fahrradfahrer zu unterscheiden. Dies liegt unter anderem daran, dass der mit nur vier Kanälen ausgestattete ibeo Lux 2010® eine zuverlässige Klassifikation erschwert. Die Segmentierung wird nicht explizit bewertet, da diese nur ein vorgelagerter Schritt für den Tracking- und Klassifizierungsalgorithmus darstellt und zudem schon im Trackingalgorithmus implementiert ist. Außerdem wird bei der verwendeten Messapparatur die Segmentzugehörigkeit der Punkte von der ECU der realen Sensoren nicht ausgegeben, wodurch ein Vergleich nicht möglich ist.

Im Abschnitt 6.4.3 werden die Verarbeitungsebenen Punktwolken und Tracking gemeinsam betrachtet. Es wird überprüft, ob eine Korrelation zwischen den Ergebnissen der Metriken der beiden Verarbeitungsebenen vorhanden ist.

6.4.1 Punktwolken

In diesem Abschnitt werden die realen und synthetischen Punktwolken bzw. die daraus generierten Occupancy Grids gegenübergestellt. Es werden verschiedene Parameter variiert, um zu überprüfen,

ob die Metriken darauf ansprechen und für die Bewertung einer Lidar-Sensorsimulation geeignet sind. Grundsätzlich werden bei der Bewertung der Simulation nur relevante Bereiche überprüft, also bspw. der Bereich um die im Sichtfeld des Sensors platzierten Fahrzeuge. Das gesamte kombinierte Sichtfeld beider Sensoren wird nicht bewertet, da dies vor allem in den realen Messdaten durch Rauschen und die dynamische Umwelt geprägt ist. In der Simulation hingegen ist die Umwelt statisch, außerdem sind Objekte wie z. B. Gras neben der Start- und Landebahn nicht detailliert abgebildet.

Dieser Abschnitt ist wie folgt aufgebaut: Im ersten Schritt wird erläutert, wie die Bodenreflexionen der simulierten Punktwolken entfernt werden, da diese die Ergebnisse verfälschen und nicht Ziel der Bewertung sind. Anschließend wird erläutert, wie aus den simulierten Punktwolken die OGs erzeugt werden, die die Basis für viele Metriken darstellen. Die OGs der realen Punktwolken werden mittels zweier unterschiedlicher Varianten berechnet, die beide nach der Erstellung der simulierten OGs erklärt werden. Daraufhin folgt die Evaluation. Der erste Schritt der Evaluation ist der Vergleich der Varianten zur Erstellung der realen OGs. Im zweiten Schritt folgt das Überprüfen des Einflusses der Anzahl der berücksichtigten Scans. Die zuvor vorgestellten Metriken werden im nächsten Schritt berechnet und deren Eignung für die Bewertung einer Lidar-Sensorsimulation bewertet. Anschließend wird der Einfluss bestimmter Parameter (Zellgröße und Abstand der Fahrzeuge zum Ego-Fahrzeug) auf die Metriken analysiert. Im letzten Schritt folgt das Manipulieren der Simulationsdaten mit dem Ziel, die Sensitivität der Metriken zu überprüfen.

Vor der Erstellung der Occupancy Grids wird in einem Preprocessing-Schritt die Bodenreflexion der Simulation entfernt. Da zum Bearbeitungszeitpunkt dieser Masterthesis noch kein Rauschmodell in der Simulation implementiert ist, erfolgt die Entfernung der Bodenreflexion nicht durch einen *Ground Removal Filter*. Stattdessen wird die Punktwolke des Szenarios mit ausschließlich der Bodenreflexion von der Punktwolke mit dem gesamten Szenario subtrahiert. Dies wird anhand der Abbildung 6-8 erklärt.

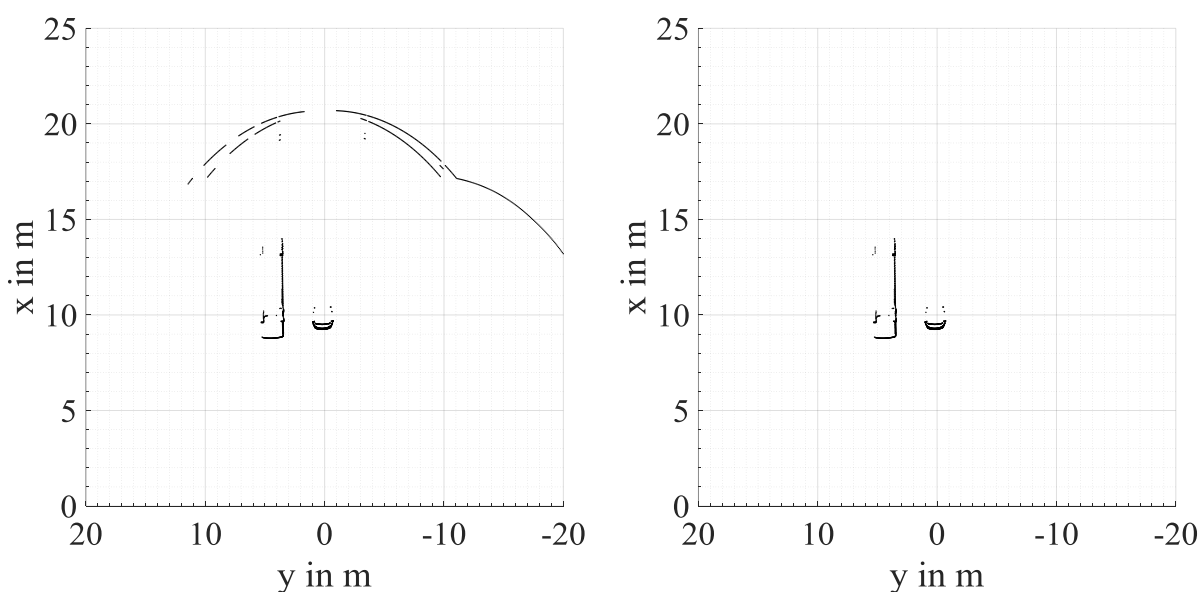


Abbildung 6-8: Vergleich simulierte Punktwolke mit und ohne Bodenreflexionen, Szenario „Stauende 2 Fahrzeuge“

Jedes simulierte Szenario wird zweifach aufgezeichnet. In der ersten Aufzeichnung wird das Szenario der Realität entsprechend nachgestellt und die Punktwolke aufgezeichnet (Abbildung 6-8 links). Anschließend werden die relevanten Fahrzeuge (außer das Ego-Fahrzeug) in der Simulation entfernt und das Szenario erneut aufgezeichnet (Abbildung 6-8 rechts). Die zweite Aufzeichnung beinhaltet die gleichen Bodenreflexionen wie das erste Szenario, jedoch keine Reflexionen der Fahrzeuge. Vor der Verarbeitung der Punktwolken wird die zweite aufgezeichnete Punktwolke (nur Bodenreflexion) von der ersten aufgezeichneten Punktwolke (Bodenreflexion und relevante Fahrzeuge) subtrahiert, indem alle Punkte in der ursprünglichen Punktwolke entfernt werden, die auch in der Punktwolke mit ausschließlich den Bodenreflexionen vorhanden sind.

Im nächsten Verarbeitungsschritt werden die Occupancy Grids (OG) erstellt. Die OGs, die aus den simulierten Punktwolken stammen, werden aus dem ersten vollständig aufgezeichneten Scan generiert. Da die Simulation statisch ist und kein Rauschmodell implementiert ist, sind alle Scans gleich und es ist unbedeutend, von welchem Scan ein Scan Grid (SG) erstellt wird. Des Weiteren spielt es keine Rolle, ob nur von einem Scan ein SG oder von mehreren Scans ein OG erstellt wird, da die Werte in den Zellen verschiedener SGs identisch sind. Die Werte in den Zellen besitzen den Wert Eins für belegt oder Null für frei. Sobald sich ein Punkt der Punktwolke innerhalb einer Zelle befindet, so ist die Zelle belegt und nimmt den Wert Eins an. Die Z-Komponente der Punkte wird nicht betrachtet, daher sind die binären OGs zweidimensional. Außerdem werden Zellen, die von anderen Zellen verdeckt sind und somit nicht im Sichtbereich des Sensors liegen, nicht als „unbekannt“ mit dem Wert 0,5 belegt, sondern als frei. Der Grund hierfür ist, dass die Berechnung eines OGs mit Berücksichtigung der Verdeckung deutlich aufwändiger ist und es für den Vergleich keine Rolle spielt, ob eine Zelle frei oder unbekannt ist, sofern das zu vergleichende OG nach dem gleichen Schema erstellt wird und ebenfalls keine Zellen mit dem Status „unbekannt“ beinhaltet.

Die Grenzen der simulierten OGs werden so gewählt, dass alle relevanten statischen Fahrzeuge in den OGs beinhaltet sind. Die Zellgröße und somit die Auflösung des OGs kann variiert werden. Da die Zellgröße einen großen Einfluss auf das Ergebnis der Metriken hat, wird der Einfluss unterschiedlicher Zellgrößen ebenfalls in diesem Abschnitt untersucht. In Abbildung 6-9 ist beispielhaft ein OG einer simulierten Punktwolke dargestellt. Das OG basiert auf der Punktwolke von Abbildung 6-8. Die Zellen in diesem Beispiel sind quadratisch mit einer Kantenlänge von 0,1 m, da bei geringer gewählter Zellgröße die Zellen visuell schwer zu erkennen sind.

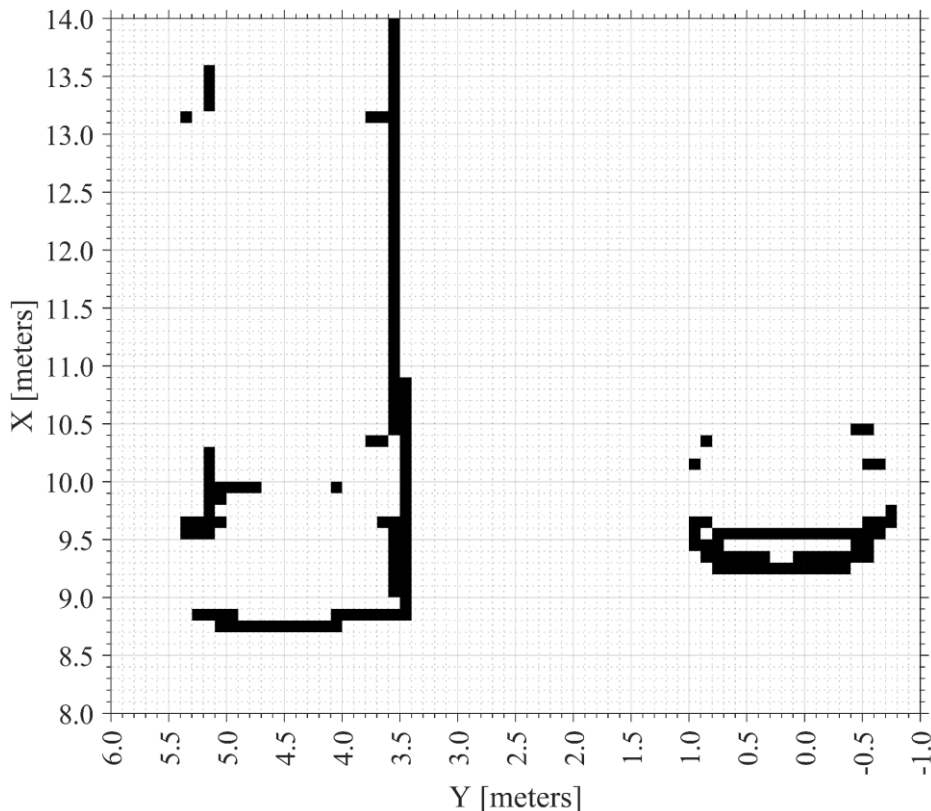


Abbildung 6-9: Beispielhaftes Occupancy Grid einer simulierten Punktwolke, Szenario „Stauende 2 Fahrzeuge“

Bei der Erstellung der Occupancy Grids aus den realen Punktwolken werden zwei Ansätze verfolgt, da durch die Berücksichtigung des Sensorrauschens nicht alle Scans identisch sind. Für die erste Variante (siehe Abbildung 6-10 oben) der Berechnung der realen OGs wird für jeden Scan ein einzelnes Scan Grid (SG) erstellt, das anschließend mit dem simulierten OG verglichen wird und bei dem für jeden Scan die Metriken einzeln berechnet werden (daher die Bezeichnung dieser Variante: „SGs einzeln“). Nachdem alle Metriken für alle Scans berechnet sind, wird pro Metrik der Mittelwert über die Scans gebildet. Die Anzahl von Scans, die berücksichtigt wird, kann variiert werden. Bei der zweiten Variante (siehe Abbildung 6-10 unten) der Berechnung der realen OGs werden alle einzelnen SGs zu einem OG zusammengefügt, indem pro Zelle der Mittelwert aller entsprechenden Zellen der einzelnen OGs gebildet wird (daher die Bezeichnung dieser Variante: „SGs gemittelt“). Das gemittelte OG besitzt probabilistische Zellwerte zwischen null und eins, die die Belegungshäufigkeit angeben. Nach der Erstellung des gemittelten OGs wird dieses mit dem simulierten OG unter Anwendung aller Metriken verglichen. Eine Gegenüberstellung der beiden Varianten zur Erstellung der realen OGs erfolgt im nächsten Absatz. Die Grenzen sowie die Auflösung der realen OGs sind identisch mit denen der vergleichbaren simulierten OGs, damit ein Vergleich möglich ist. In Abbildung 6-11 ist ein auf realen Daten basierendes gemitteltes OG dargestellt (Variante „SGs gemittelt“), welches das gleiche Szenario wie das simulierte OG in Abbildung 6-9 zeigt. Die Zellauflösung beträgt ebenfalls 0,1 m und der Mittelwert wird über 250 Scans gebildet. Je dunkler eine Zelle eingefärbt ist, desto häufiger ist die Zelle in den 250 Scans belegt (weiß entspricht 100 % frei, schwarz entspricht 100 % belegt).

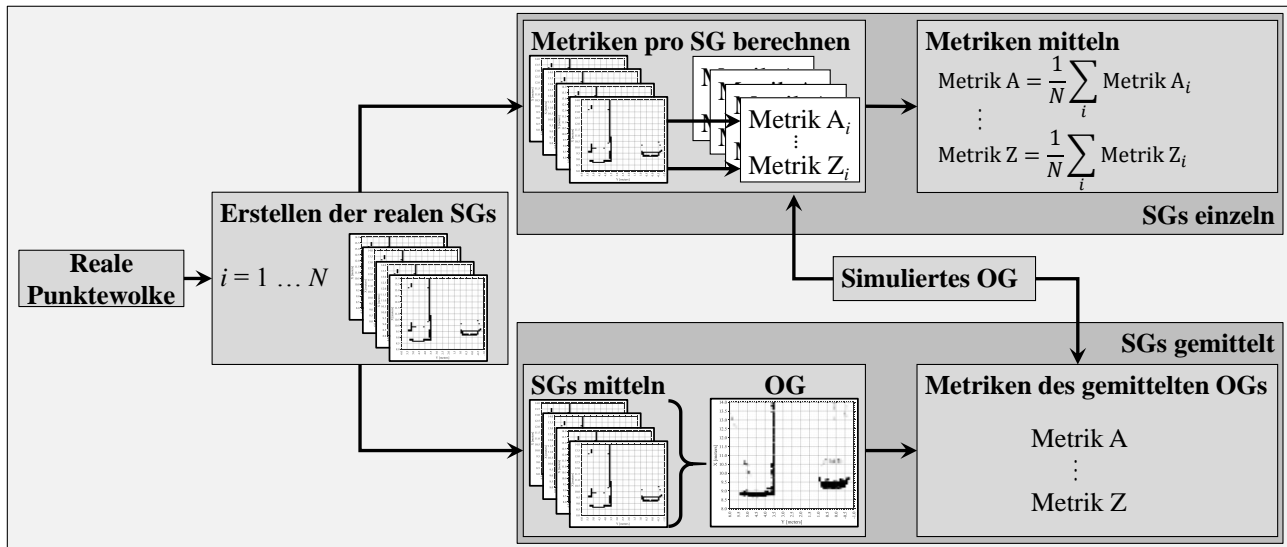


Abbildung 6-10: Erstellung der verschiedenen OG-Varianten

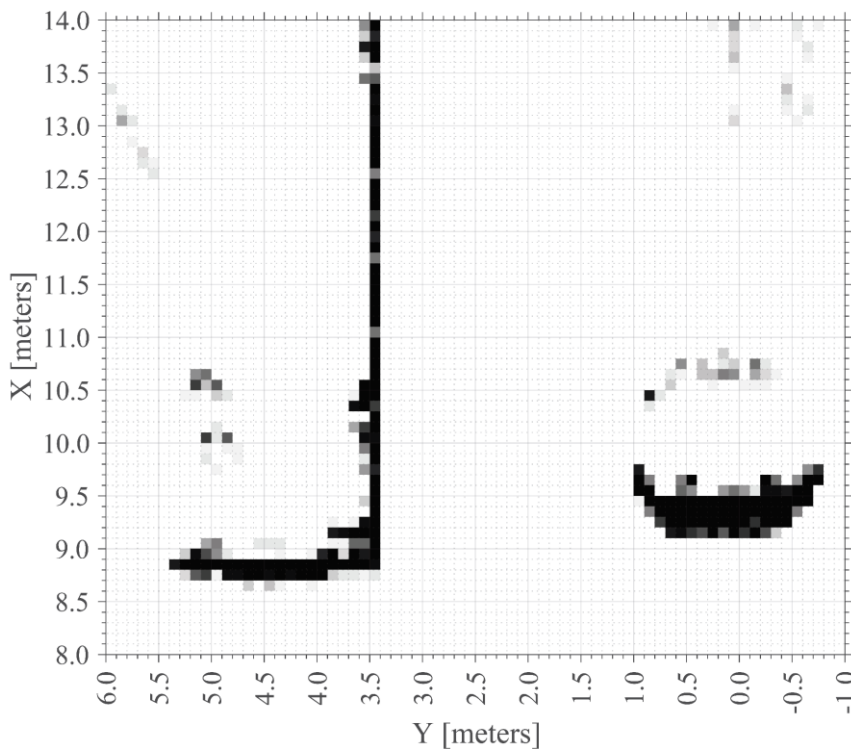


Abbildung 6-11: Beispielhaftes gemitteltes Occupancy Grid einer realen Punktwolke, Szenario „Stauende 2 Fahrzeuge“

Im ersten Schritt der Evaluation werden die zuvor erklärten Varianten zur Erstellung der realen OGs verglichen. Die gewählten Metriken sind für diesen Versuch zunächst nicht von Bedeutung und werden im späteren Verlauf dieses Abschnitts ausführlich erklärt. Die Anzahl der betrachteten Scans beträgt für den Vergleich im Folgenden 250, die Zellauflösung wird in diesem Beispiel auf 0,04 m festgelegt. In Abbildung 6-12 ist das Ergebnis ausgewählter Metriken für das Szenario „Stauende 2

Fahrzeuge“ mit 10 m Abstand der Fahrzeuge zum Ego-Fahrzeug und den genannten Randbedingungen dargestellt. Die schwarzen (dunkleren) Balken repräsentieren die Ergebnisse der Variante mit der Mittelung der Metriken über die SGs (Variante „SGs einzeln“), während die hellgrauen (helleren) Balken das Ergebnis der Variante der gemittelten SGs (Variante „SGs gemittelt“) darstellen. An diesem Beispiel ist zu erkennen, dass die zweite Variante bessere Ergebnisse, also höhere Übereinstimmung der realen und simulierten OGs, erzielt. Diese Feststellung ist auch bei anderen Zellauflösungen und anderen Szenarien zu beobachten. Der Grund hierfür ist, dass durch das Aufsummieren der 250 einzelnen SGs zu einem probabilistischen OG deutlich mehr Zellen belegt sind, als bei einem einzelnen SG. Dies erhöht zum einen die Wahrscheinlichkeit, dass eine simulierte Zelle mit einer realen Zelle übereinstimmt, und zum anderen ist die Anzahl der belegten Zellen höher, die bei vielen Metriken zur Normierung herangezogen wird.

Die probabilistischen realen OGs spiegeln die Realität besser als OGs aus einzelnen Scans wider, da die reale Punktwolke auch mit Rauschen versehen ist und die Betrachtung über mehrere Scans dies berücksichtigt. Für die weitere Evaluation werden daher die simulierten OGs nur mit den probabilistischen realen OGs (Variante „SGs gemittelt“) verglichen.

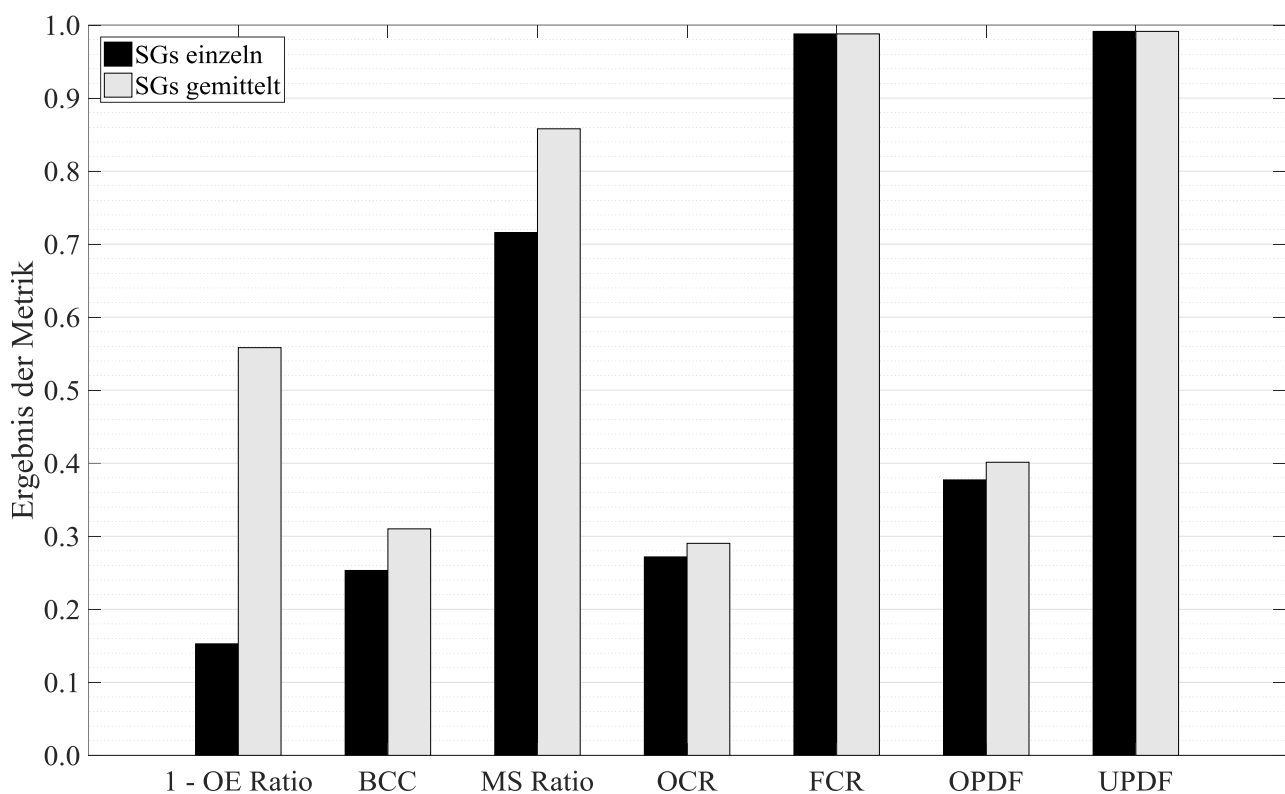


Abbildung 6-12: Vergleich der Varianten zur Erstellung der realen OGs, Szenario „Stauende 2 Fahrzeuge“, 250 Scans, Zellauflösung 0,04 m

Als nächstes wird der Einfluss der Anzahl der Scans, die für die Berechnung berücksichtigt werden, untersucht. Es wird ermittelt, ob auch weniger als 250 berücksichtigte Scans vergleichbare Ergebnisse erzielen, da dies Rechenzeit einspart. Für diese Untersuchung wird das gleiche Szenario mit den gleichen Randbedingungen wie bei der vorherigen Untersuchung gewählt („Stauende 2 Fahrzeuge“ mit

10 m Abstand der Fahrzeuge zum Ego-Fahrzeug und Zellauflösung 0,04 m). In Abbildung 6-13 sind ausgewählte Metriken in Abhängigkeit der Anzahl der berücksichtigten Scans dargestellt. Die Metriken steigen bis auf wenige Ausnahmen mit der Anzahl der Scans, wobei die Veränderung zwischen 20 und 100 berücksichtigten Scans am deutlichsten ausfällt. Ein Grund hierfür ist, dass die Wahrscheinlichkeiten in den Zellen des OGs, welches aus 20 einzelnen SGs gemittelt ist, noch ungenau sind. Mit steigender Anzahl der berücksichtigten Scans steigt die Belegungsgenauigkeit und die Abweichungen zum simulierten OG werden geringer. Ein weiterer Grund für das Ansteigen der Metriken mit steigender Anzahl der berücksichtigten Scans ist, dass einige Metriken mit der Anzahl der belegten Zellen normiert sind und diese Anzahl ebenfalls mit steigender Anzahl der berücksichtigten Scans zunimmt. Als Folge dessen steigen auch die Metriken.

Da der Einfluss der Anzahl der berücksichtigten Scans nicht zu vernachlässigen ist, werden die nachfolgenden Berechnungen mit der Maximalanzahl der Scans durchgeführt. Diese wird für alle Szenarien auf 250 festgelegt und entspricht einer Aufzeichnungsdauer von 20 Sekunden.

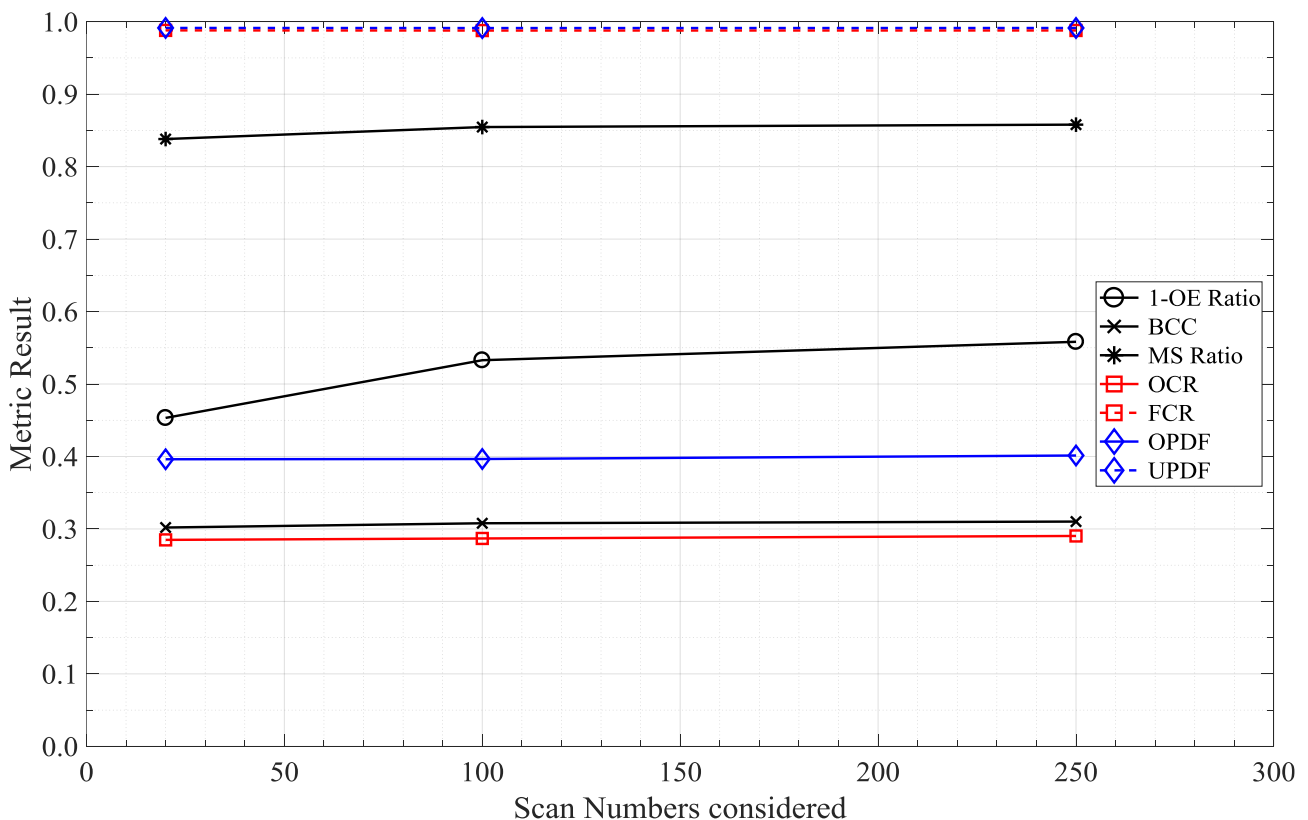


Abbildung 6-13: Einfluss der Anzahl der berücksichtigten Scans, Szenario „Stauende 2 Fahrzeuge“, 250 Scans, Zellauflösung 0,04 m

Im Folgenden werden die in Abschnitt 5.2 vorgestellten Punktwolken-Metriken (siehe Tabelle 5-2) bei den unterschiedlichen Szenarien für den Vergleich der Simulation mit den realen Messdaten angewendet und deren Ergebnisse analysiert. Außerdem wird eingeordnet, ob die vorgestellten Metriken für eine weitere Anwendung in zukünftigen Bewertungen von Lidar-Sensormodellen geeignet

sind. Am Ende der Beschreibung der einzelnen Metriken befindet sich eine Tabelle, die alle Ergebnisse für das Szenario „Stauende 2 Fahrzeuge“ zusammenfasst (siehe Tabelle 6-5 auf Seite 77). Die Tabellen für die anderen vier statischen Szenarien befinden sich in Anhang A.1.

Overall Error

Der Overall Error (OE) wird nur für die OGs berechnet und nicht für die Punktwolken, da dies die Berechnung der Euklidischen Abstands-Matrix erfordert und die hierfür benötigte Zuordnung der Reflexionen zu den Kanälen in der simulierten Punktwolke nicht bekannt ist. Um den Overall Error zu normieren, wird das Ergebnis mit der Anzahl der belegten Zellen beider OGs dividiert. Außerdem wird zur besseren Verständlichkeit und zur Vorbeugung von Fehlinterpretationen beim Vergleich mit anderen Metriken der normierte Overall Error (OE Ratio) von Eins subtrahiert (1-OE Ratio). Somit erzielen zwei identische OGs den Wert Eins, während zwei OGs ohne Übereinstimmung den Wert Null erzielen. Der Match Score entspricht der Berechnung des Overall Errors bis auf das Quadrieren anstelle vom Betrag innerhalb der Summation und wird daher nicht berechnet.

Abbildung 6-14 zeigt die Metrik 1-OE Ratio in Rot (helle durchgezogene Linie, linke Ordinatenachse) für die fünf verschiedenen Szenarien in Abhängigkeit des Abstandes der Fahrzeuge vom Ego-Fahrzeug. Da beim Szenario „parkende Autos“ die Fahrzeuge in unterschiedlichen Distanzen zum Ego-Fahrzeug stehen, ist der Wert 1-OE Ratio für dieses Szenario beim Abstand 1 m mit einem roten (hellen) \times markiert. Die meisten Metriken nehmen Werte zwischen 0,53 und 0,66 an. Auffällig sind die deutlich niedrigeren Werte bei den Abständen 40 und 50 m der Szenarien „i3 statisch Front“ und „i3 statisch Heck“. Ein Grund hierfür ist, dass die Anzahl der belegten Zellen mit zunehmendem Abstand deutlich abnimmt. Dies führt zum einen dazu, dass sich die Wahrscheinlichkeit, mit der zwei Zellen vom realen und simulierten OG den gleichen Wert beinhalten, erheblich verringert. Zum anderen sinken die Belegungswahrscheinlichkeiten des probabilistischen realen OGs, wodurch die Abweichung zum binären simulierten OG größer ist. Die Szenarien „Stauende 2 Fahrzeuge“ und „i3 statisch Seite“ sind nicht von einer deutlich geringeren Anzahl der belegten Zellen betroffen, da hier auch bereits bei näheren Abständen mehr belegte Zellen existieren und in größeren Entfernungen genug Zellen vorhanden sind. Dadurch wirkt sich die Verringerung der Anzahl der Zellen nicht auf die Ergebnisse der Metrik der beiden Szenarien aus.

Das Verwenden der Overall Error-Metrik zeigt, dass die ausgegebenen Absolutwerte (OE) zunächst schwierig zu interpretieren sind. Die eingeführte Normierung erleichtert das Vergleichen verschiedener Simulationen. Die OE-Metrik (OE, OE Ratio, 1-OE Ratio) ist demnach für die Bewertung von Lidar-Sensorsimulationen nicht ungeeignet.

BCC und PCC

Der Kreuzkorrelationskoeffizient nach Baron (BCC) sowie der Korrelationskoeffizient nach Pearson (PCC) erzielen identische Ergebnisse. Dies steht im Widerspruch zu den von Schärmann et al. produzierten Ergebnissen, konnte jedoch trotz Kontakt mit Khelifa Baizid¹¹⁷, der ebenfalls beide Koef-

¹¹⁷ Vgl. Baizid, K.: Barons and Pearson Cross Correlation Coefficient (2018).

fizienten berechnet hat, nicht gelöst werden. Für die folgenden Vergleiche wird nur der Kreuzkorrelationskoeffizient nach Baron aufgelistet. In Abbildung 6-14 ist der BCC in Schwarz (dunkle abgesetzte Linie, rechte Ordinatenachse) für die fünf verschiedenen Szenarien in Abhängigkeit des Abstandes der Fahrzeuge vom Ego-Fahrzeug dargestellt. Auch hier ist der BCC für das Szenario „parkende Autos“ bei der Distanz 1 m zu finden. Der BCC wird mit steigendem Abstand der Fahrzeuge grundsätzlich geringer, wobei bei 40 und 50 m Distanz fast keine Korrelation zu sehen ist. Der Ausreißer mit negativer Korrelation beim Szenario „i3 statisch Heck“ ist auf eine fehlerhafte Positionierung des BMW i3 zurückzuführen. Da bei diesem Versuch kein ADMA im i3 verbaut ist, liegt möglicherweise ein Messfehler bei der Messung der Distanz mit dem Maßband vor.

Der Barons Kreuzkorrelations-Koeffizient (bzw. Pearsons Korrelations-Koeffizient) vergleicht die OGs auf Zellenbasis und ist daher sehr empfindlich bei geringfügigen Verschiebungen der OGs. Daher ist auch die Abstandsabhängigkeit des BCCs ausgeprägt. Vor Anwendung der BCC-Metrik muss sichergestellt werden, dass die relevanten Objekte in der Realität und Simulation mit bestmöglicher Genauigkeit die gleichen Positionen einnehmen. Insgesamt liefert die BCC-Metrik jedoch Ergebnisse, die eine Bewertung der Lidar-Sensorsimulation ermöglichen.

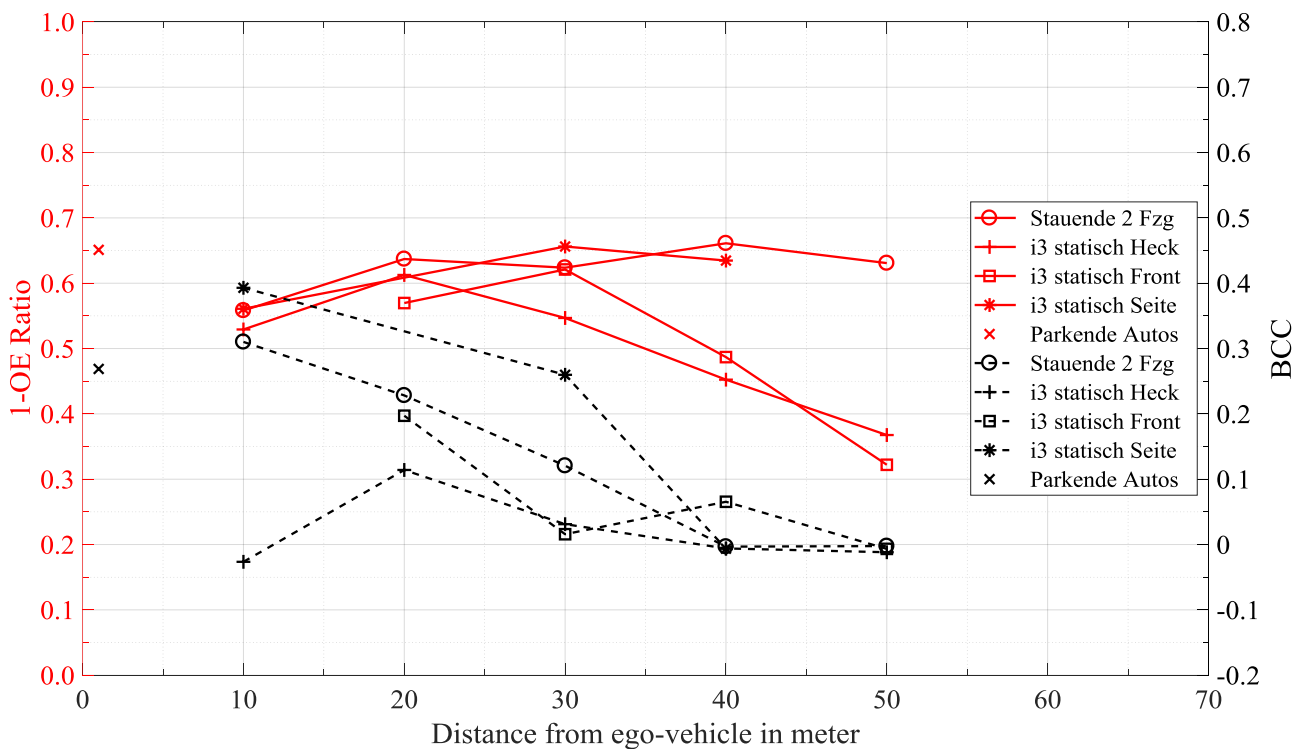


Abbildung 6-14: 1-OE Ratio und BCC in Abhängigkeit des Abstandes für verschiedene Szenarien

Map Score

Der Map Score (MS) sowie der Map Score für belegte Zellen (MS OCC) ist für die Lidar-Sensorsimulation nicht direkt anwendbar, da das simulierte OG nur die Werte Eins oder Null in den Zellen beinhaltet. Dies führt dazu, dass der Map Score den Wert $-\infty$ annimmt, sobald ein Zellenpaar des realen und simulierten OGs gegensätzliche Werte besitzt, also die Kombination $m_{\text{real}} = 1$ und

$m_{\text{simuliert}} = 0$ bzw. umgekehrt. Da dies häufig der Fall ist, wird der Map Score nach dem Vorschlag von Fernández-Maridgal¹¹⁸ so abgewandelt, dass die Zellwerte des simulierten OGs entweder 0,0001 für frei oder 0,9999 für belegt annehmen. Der Map Score wird mit der Anzahl aller Zellen eines OGs normiert (MS Ratio) und der Map Score für belegte Zellen wird mit der Anzahl der belegten Zellen beider OGs normiert (MS Ratio OCC).

Die MS-Metrik gibt ähnlich wie der Overall Error einen Absolutwert aus, der schwierig zu interpretieren ist. Eine Normierung erleichtert auch hier den Vergleich verschiedener Simulationen. Da beim MS alle Zellen berücksichtigt werden, fällt dieser tendenziell besser aus, wenn viele freie Zellen vorhanden sind. Insgesamt ist die MS-Metrik (MS und MS Ratio) für die Bewertung einer Lidar-Sensorsimulation nicht ungeeignet. Die Berechnung des Map Scores für belegte Zellen (MS OCC und MS Ratio OCC) ist zwar zunächst sinnvoll, das Ergebnis ist jedoch schwierig zu interpretieren. Dies liegt daran, dass eine Abweichung zweier Zellen von größer als 0,5 einen negativen MS OCC verursacht. Da dies bei vielen Zellpaaren der Fall ist, fällt der MS OCC oft negativ aus. Durch die Normierung ist die Interpretation des MS OCC einfacher und verschiedene Simulationen vergleichbar, jedoch wird trotzdem von dieser Metrik (MS OCC und MS Ratio OCC) abgeraten.

Bhattacharyya-Entfernung

Die Bhattacharyya-Entfernung wird nicht berechnet, da im Sensormodell zum jetzigen Stand kein Rauschen implementiert ist und die Histogramme der Simulation im Idealfall bei Positionierung einer ebenen Fläche vor dem Lidar-Sensor nur eine Klasse aufweisen. Außerdem ist das Ergebnis des Vergleichs von Histogrammen abhängig von der gewählten Klassenbreite und somit nicht vergleichbar.

Punktewolke-zu-Punktewolke-Distanz

Die Punktewolke-zu-Punktewolke-Distanz (PC2PC-Distanz) wird für alle jeweils 250 Scans pro Szenario berechnet. Das Ergebnis ist jedoch nicht normiert und ein Vergleich gleicher Szenarien bei unterschiedlichen Abständen der Fahrzeuge schwierig, da die Anzahl der Punkte in der Punktewolke einen großen Einfluss auf das Ergebnis der Punktewolke-zu-Punktewolke-Distanz besitzt und die Anzahl der Punkte mit dem Abstand variiert.

Hit/Miss-Vergleich

Der Hit/Miss-Vergleich wird nicht durchgeführt, da keine Informationen über die Kanalzugehörigkeit der reflektierten Punkte in der simulierten Punktewolke vorliegen.

OCR/FCR und OPDF/UPDF

Das Occupied Cells Ratio (OCR) und Free Cells Ratio (FCR) sowie die Occupied Picture-Distance-Function (OPDF) und die Unoccupied Picture-Distance-Function (UPDF) werden den Formeln entsprechend berechnet. Wichtig ist bei diesen Metriken, dass diese immer im Verbund (occupied/free bzw. occupied/unoccupied) betrachtet werden, da eine Verbesserung der einen Metrik häufig eine Verschlechterung der anderen Metrik mit sich bringt. Außerdem ist bspw. das $\text{OCR} = 1$, wenn allen

¹¹⁸ Fernández-Madriral, J.-A.: E-Mail-Verkehr bezüglich Map Score (2018).

belegten Zellen der Realdaten eine belegte Zelle der Simulationsdaten zugewiesen ist. Wenn die Simulation jedoch zusätzlich freien realen Zellen den Status „belegt“ zuweist, so bleibt weiterhin $OCR = 1$, während das FCR sinkt. Die Berechnung dieser vier Metriken erfordert eine Definition, die besagt, wann eine Zelle als leer und wann als belegt bezeichnet wird. Mit der Definition einer leeren Zelle bei einem Zellwert $m_i < 0,5$ und einer belegten Zelle bei einem Zellwert $m_i \geq 0,5$ errechnet sich bei größeren Distanzen der Fahrzeuge ein OCR und OPDF von Null, da nur in wenigen Fällen beide Zellwerte des realen und simulierten OGs oberhalb des Schwellwertes für eine belegte Zelle liegen. Das FCR und die UPDF sind hingegen meistens knapp unterhalb des Wertes 1,0, da sehr viele freie Zellen als frei simuliert werden und allgemein deutlich mehr freie Zellen existieren als belegte.

Ein Parameter der OPDF bzw. UPDF ist der Suchradius, in dem nach belegten bzw. freien Zellen im jeweils anderen OG gesucht wird. Baizid et al. geben keinen expliziten Wert an. In einem ersten Versuch werden drei verschiedene Radien ausprobiert, siehe Abbildung 6-15. Der Radius wird in Anzahl der Auflösungszellen angegeben und beträgt 1, 2 oder 3, somit wird in den umliegenden 9, 25 oder 49 Zellen nach einer belegten bzw. freien Zelle gesucht. Wie zu erwarten steigen die OPDF- und UPDF-Metriken mit zunehmenden Radius, da die Wahrscheinlichkeit steigt, dass sich innerhalb des Radius eine belegte bzw. freie Zelle befindet. Falls nicht explizit angegeben, beträgt der Suchradius in dieser Ausarbeitung eins. Eine mögliche Erweiterung der OPDF- und UPDF-Metrik stellt die Anpassung des Suchradius in Abhängigkeit des Abstandes der Zelle vom Sensor dar.

Die Metriken OCR, FCR, OPDF und UPDF basieren wie der BCC auf einem zellweisen Vergleich der OGs und sind daher auch sensitiv gegenüber einer Abstandsänderung der Fahrzeuge, wobei sich die zusammengehörigen Metriken (OCR und FCR sowie OPDF und UPDF) immer konträr verhalten. Speziell die OCR- und OPDF-Metriken, die sich auf belegte Zellen beziehen, zeigen jedoch, wie genau die Zellen des simulierten OGs mit den Zellen des realen OGs übereinstimmen. Daher sind alle vier Metriken für die Bewertung einer Lidar-Sensorsimulation nicht ungeeignet.

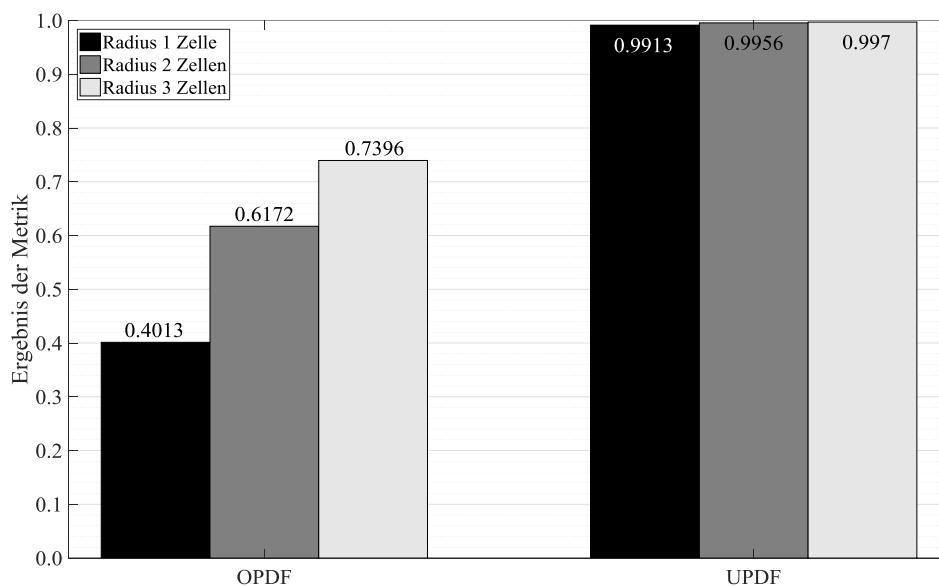


Abbildung 6-15: Variation des Suchradius der OPDF- und UPDF-Metrik, Szenario „Stauende 2 Fahrzeuge“, Abstand 10 m, 250 Scans, Zellauflösung 0,04 m

Hausdorff-Distanz

Die Hausdorff-Distanz zwischen den realen und simulierten Punktwolken wird zusätzlich zu den in Abschnitt 5.2 vorgestellten Metriken berechnet. Sie wird analog zu der Gleichung (5-18), die für die Bewertung des Trackings bestimmt ist, berechnet. Als Metrik $d(\mathbf{x}, \mathbf{y})$ wird der euklidische Abstand verwendet. Für jeden Punkt einer Punktwolke wird der nächstgelegene Punkt der anderen Punktwolke gesucht und der euklidische Abstand berechnet. Die Hausdorff-Metrik ist das Maximum aller berechneten Abstände. Da die Metrik nicht symmetrisch ist, wird die gleiche Berechnung für die andere Punktwolke durchgeführt und das Maximum der beiden zuvor ermittelten Hausdorff-Metriken als gesamte Hausdorff-Metrik festgelegt. Da die Hausdorff-Metrik jeden Punkt berücksichtigt, ist die Metrik häufig durch Ausreißer bestimmt. Von der Anwendung zur Bewertung einer Lidar-Sensorsimulation wird daher abgeraten.

Vergleich der Abstände der Schwerpunkte

Eine weitere Metrik, die direkt auf die Punktwolken angewendet wird, ist der Vergleich des euklidischen Abstands der Schwerpunkte der beiden Punktwolken. Dieser ermöglicht eine einfache erste Abschätzung, wie ähnlich die Punktwolken sind.

Die Tabelle 6-5 beinhaltet alle berechneten Metriken für das Szenario „Stauende 2 Fahrzeuge“. Die Zellauflösung beträgt 0,04 m bei allen Abständen. Für jede Metrik ist das beste erreichte Ergebnis fett markiert. Auffällig ist, dass die Metriken, die einen zellweisen Vergleich durchführen und die sich auf die belegten Zellen beschränken, hauptsächlich bei geringen Abständen die besten Werte erzielen. Umgekehrt hingegen erzielen die Metriken, die alle Zellen in die Berechnung mit einbeziehen (und dadurch hauptsächlich freie Zellen), bei größeren Abständen bessere Ergebnisse. Die Ergebnisse der OPDF und UPDF sind für alle drei Suchradien angegeben. Die analog generierten Ergebnisse in tabellarischer Form für die anderen vier Szenarien befinden sich im Anhang A.1. Auch bei diesen beträgt die Zellgröße 0,04 m für alle Abstände.

Tabelle 6-5: Metriken für das Szenario „Stauende 2 Fahrzeuge“ bei verschiedenen Abständen, Zellauflösung 0,04 m

Metrik	Abstand der Fahrzeuge zum Ego-Fahrzeug in Metern				
	10	20	30	40	50
OE	606,500	356,760	168,568	119,960	106,332
OE Ratio	0,442	0,363	0,376	0,339	0,369
1-OE Ratio	0,558	0,637	0,624	0,661	0,631
PCC	0,310	0,228	0,121	-0,003	-0,002
BCC	0,310	0,228	0,121	-0,003	-0,002
MS	22520,616	24181,902	24930,539	25273,283	25310,229
MS Ratio	0,858	0,921	0,950	0,963	0,964
MS OCC	-2889,232	-1566,164	-1103,598	-823,087	-805,083
MS Ratio OCC	-4,206	-4,449	-7,166	-7,839	-8,945
OCR	0,290	0,213	0,119	0,000	0,000
FCR	0,988	0,993	0,996	0,997	0,997
OPDF ($r = 1$)	0,401	0,319	0,175	0,008	0,000
OPDF ($r = 2$)	0,617	0,516	0,347	0,107	0,000
OPDF ($r = 3$)	0,740	0,646	0,505	0,287	0,024
UPDF ($r = 1$)	0,991	0,995	0,997	0,998	0,998
UPDF ($r = 2$)	0,996	0,998	0,998	0,999	0,999
UPDF ($r = 3$)	0,997	0,998	0,999	0,999	0,999
Hausdorff-Distanz in m	3,290	1,537	2,468	2,925	4,290
Schwerpunkt-Distanz in m	0,138	0,175	0,410	0,550	0,576
PC2PC-Distanz in m	90,015	50,924	56,759	66,428	65,482

Mit den folgenden zwei Untersuchungen wird überprüft, welchen Einfluss verschiedene Parameter auf die Ergebnisse der Metriken haben.

Zunächst wird die Zellauflösung der OGs analog zu Grewe et al.¹¹⁹ variiert. Es wird überprüft, wie sich die Metriken bei unterschiedlichen Zellauflösungen verhalten. In der bisherigen Auswertung wird eine Zellauflösung von 0,04 m verwendet mit der Begründung, dass die Distanzauflösung des ibeo Lux 2010® ebenfalls 0,04 m beträgt. Im Folgenden werden die Auflösungen 0,05 m, 0,10 m, 0,125 m, 0,2 m, 0,25 m und 0,5 m untersucht. Abbildung 6-16 zeigt das Ergebnis einiger Metriken beim Szenario „Stauende 2 Fahrzeuge“ im Abstand von 10 Metern, wobei die Zellauflösung auf der Abszissenachse aufgetragen ist. Metriken wie das FCR und die UPDF werden nur von den freien Zellen beeinflusst und sind nicht stark von der Zellauflösung abhängig, da die Anzahl der freien Zellen bei allen Auflösungen deutlich größer ist als die Anzahl der belegten Zellen. Im Gegensatz hierzu sprechen die konträren Metriken OCR und OPDF auf eine Steigerung der Zellgröße mit besseren

¹¹⁹ Vgl. Grewe, R. et al.: Accuracy of an Automotive Occupancy Grid (2012), S. 23.

Ergebnissen an. Ähnlich verhält sich der BCC, der sich ebenfalls mit steigender Zellgröße verbessert. Die Metrik 1-OE Ratio besitzt keine ausgeprägte Abhängigkeit der Zellauflösung. Das beschriebene Verhalten der Metriken ist auch in den anderen Szenarien zu beobachten.

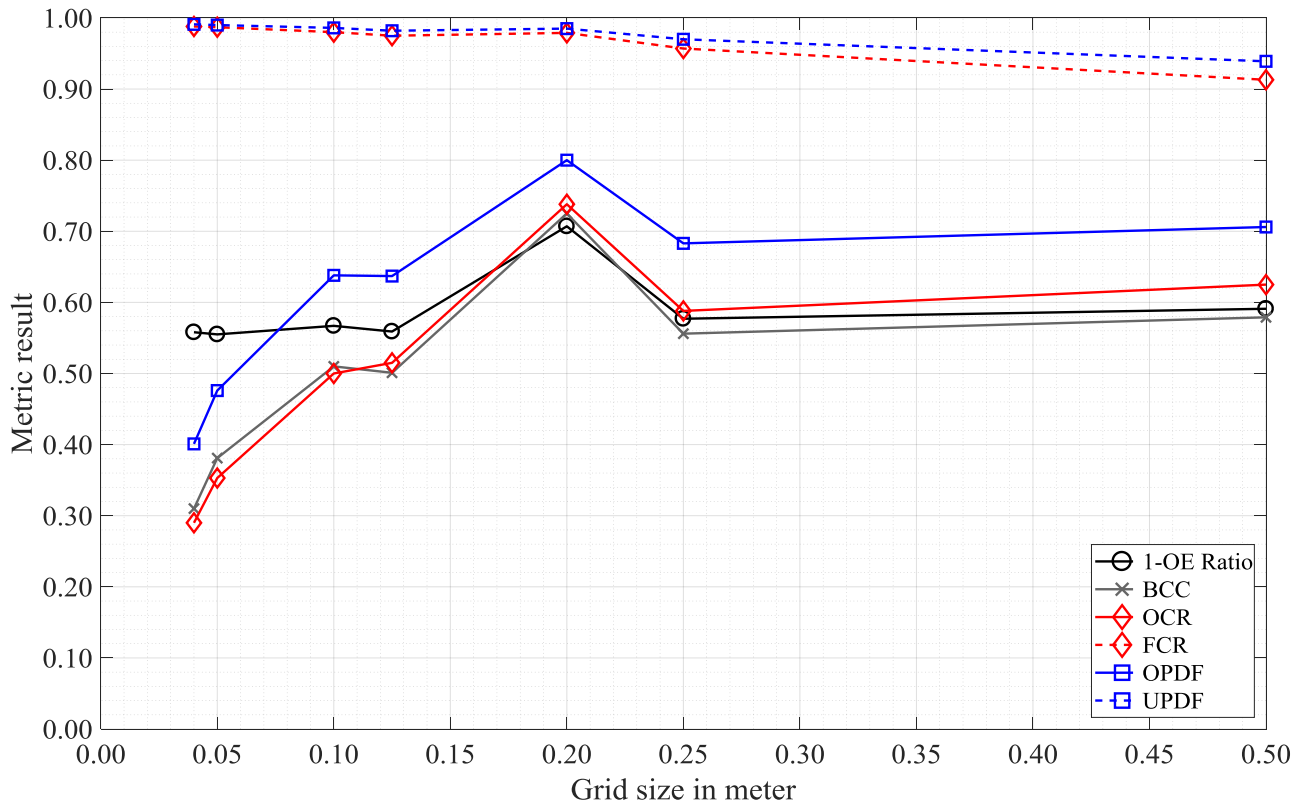


Abbildung 6-16: Variation der Zellauflösung, Szenario „Stauende 2 Fahrzeuge“, Abstand 10 m

Abgesehen von der FCR- und UPDF-Metrik sind die Ergebnisse der Metriken bei der Zellauflösung 0,2 m deutlich besser als die Ergebnisse der anderen Zellauflösungen. Der Grund hierfür wird beim Betrachten der zugehörigen OGs in Abbildung 6-17 gefunden. Die oberen beiden Diagramme stellen das reale und simulierte OG mit der Zellgröße 0,2 m dar. Zum Vergleich ist in den unteren beiden Diagrammen das reale und simulierte OG mit der Zellgröße 0,25 m abgebildet. Es ist ersichtlich, dass viele Zellen der rechten Seite des linken Fahrzeuges bei 0,2 m Zellauflösung übereinstimmen, während bei 0,25 m Zellauflösung die gleichen Zellen zwischen realen und simulierten Daten um eine Auflösungsspalte verschoben sind. Dieses Beispiel verdeutlicht, dass es wichtig ist, mehrere Zellauflösungen zu betrachten, um Fehler durch Unregelmäßigkeiten ausschließen zu können. Außerdem zeigt sich, dass die Metriken sensitiv gegenüber einer Verschiebung der Zellen sind und tendenziell korrekt darauf ansprechen.

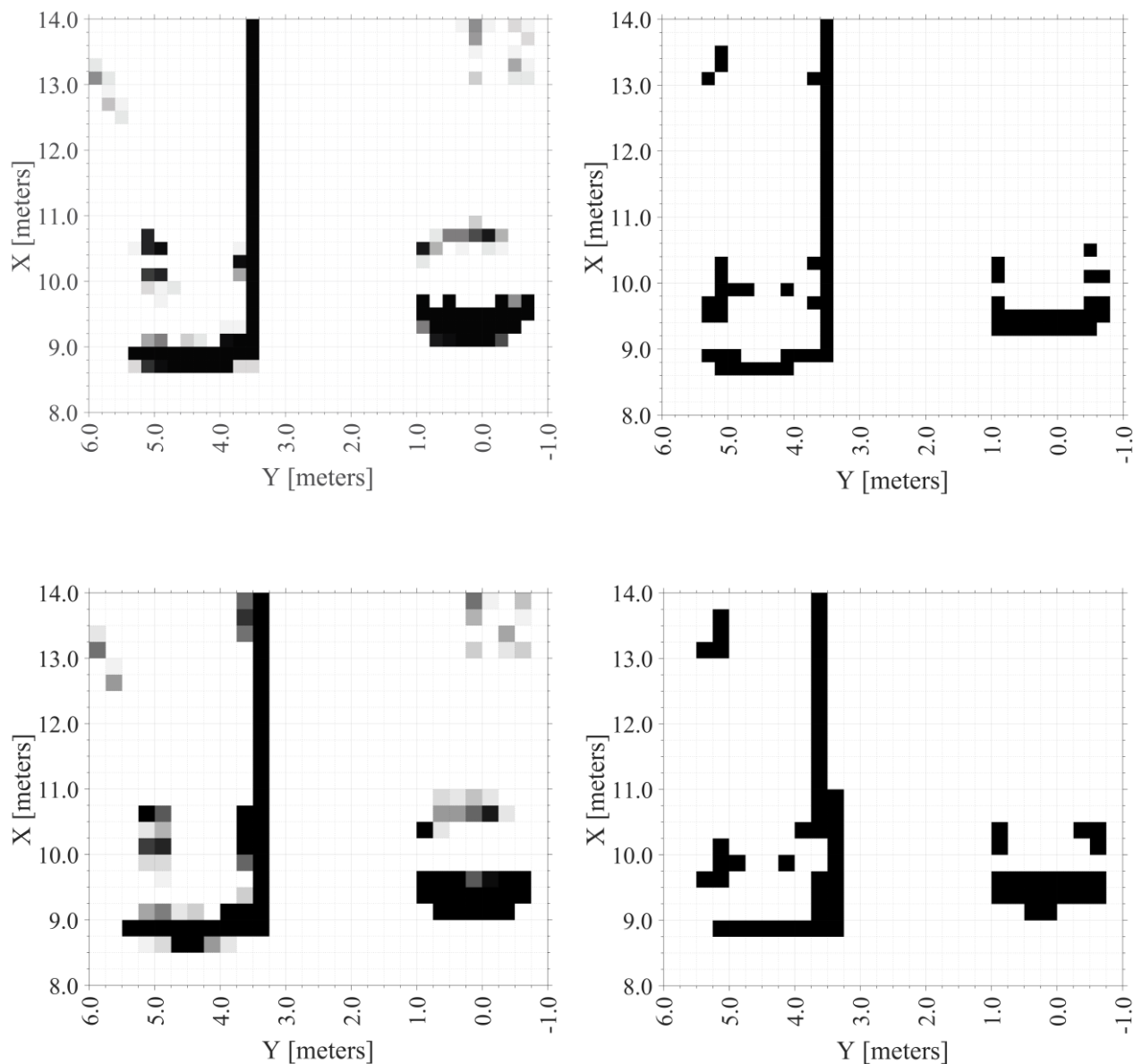


Abbildung 6-17: OGs für den Vergleich der Zellauflösung, Szenario „Stauende 2 Fahrzeuge“, Abstand 10 m. Oben links: reales OG mit Zellgröße 0,2 m, oben rechts: simuliertes OG mit Zellgröße 0,2 m, unten links: reales OG mit Zellgröße 0,25 m, unten rechts: simuliertes OG mit Zellgröße 0,25 m

Ein weiterer Teil der Evaluation ist die Untersuchung, wie sich die Ergebnisse der Metriken bei den gleichen Szenarien mit unterschiedlichen Abständen verhalten. Hierzu wird beispielhaft die BCC-Metrik betrachtet. In Abbildung 6-18 ist das Ergebnis des BCC in Abhängigkeit der Zellauflösung und des Abstandes der Fahrzeuge zum Ego-Fahrzeug beim Szenario „Stauende 2 Fahrzeuge“ dargestellt. Es sind zwei Tendenzen zu erkennen: Erstens steigt das Ergebnis der Metrik mit zunehmender Zellgröße (wie bereits Abbildung 6-16 zeigt) und dies gilt unabhängig vom betrachteten Abstand der Fahrzeuge. Zweitens sinkt das Ergebnis der Metrik bei konstanter Zellauflösung mit steigendem Abstand. Die zweite Tendenz ist hauptsächlich bei den kleineren Zellgrößen ($\leq 0,2$ m) zu beobachten. Dies liegt daran, dass die Wahrscheinlichkeit, mit der zwei Zellen vom realen und simulierten OG den gleichen Wert beinhalten, bei größeren Abständen aufgrund der Anzahl der Reflexionspunkte

deutlich geringer ist als bei kleineren Abständen. Bei größerer Zellgröße fällt dieser Effekt nicht so stark ins Gewicht, da auf das gesamte Fahrzeug nur wenige Zellen fallen und die Reflexionspunkte unter diesen wenigen Zellen aufgeteilt werden.

Bei anderen Metriken, die keine ausgeprägte Abhängigkeit von der Zellauflösung aufweisen, wie z. B. der 1-OE Ratio-Metrik, ist keine ausgeprägte Abstandsabhängigkeit vorhanden.

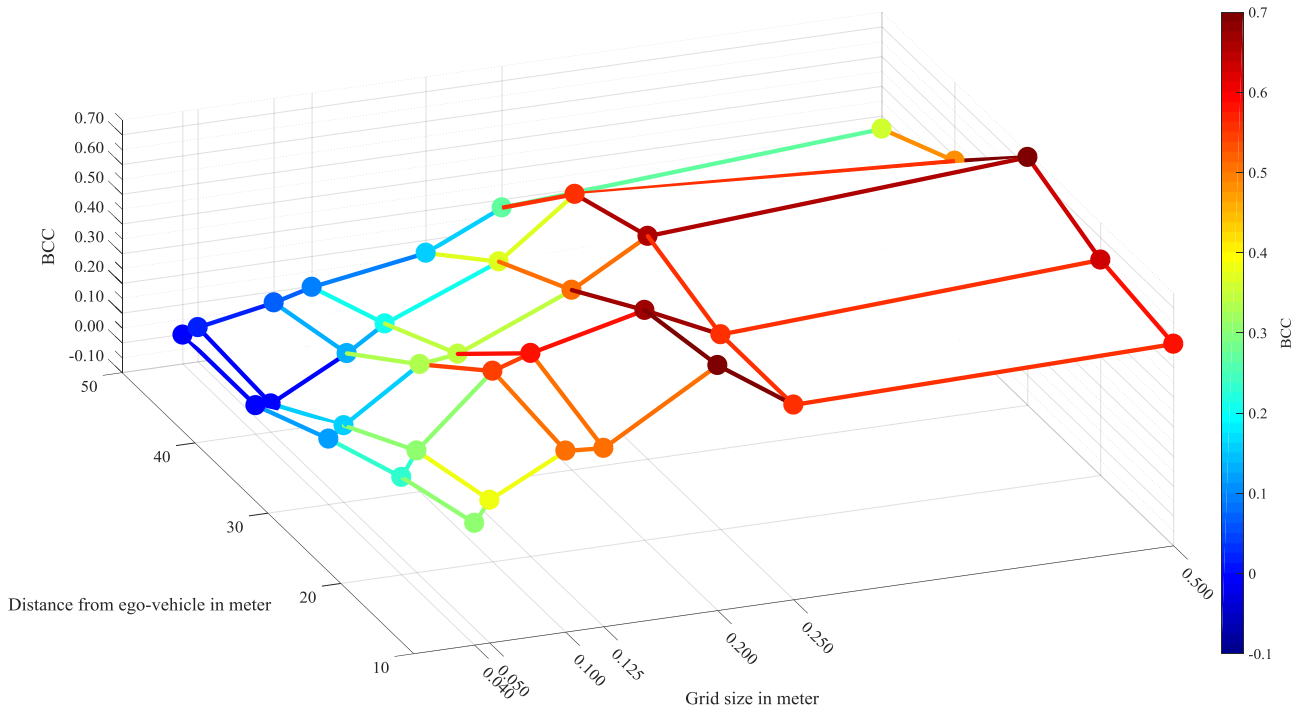


Abbildung 6-18: Variation der Zellauflösung und des Abstandes für die BCC-Metrik, Szenario „Stauende 2 Fahrzeuge“

Abschließend wird noch die Sensitivität der Metriken durch Manipulation der Punktwolke überprüft. In der simulierten Punktwolke des Szenarios „Stauende 2 Fahrzeuge“ für 10 m Abstand werden die Reflexionspunkte des BMW i3 manuell entfernt und daraufhin die Metriken erneut berechnet. Das Ziel dieser Untersuchung ist zu überprüfen, ob die Metriken auf das geänderte Szenario mit schlechteren Ergebnissen ansprechen. Das Balkendiagramm in Abbildung 6-19 zeigt den Vergleich zwischen den Metriken mit Berücksichtigung des i3 (schwarz, dunkel) und ohne Berücksichtigung des i3 (grau, hell). Ohne Berücksichtigung des i3 erzielen die BCC-, OCR- und OPDF-Metriken schlechtere Ergebnisse. Wie zu erwarten steigen außerdem die Metriken FCR und UPDF geringfügig, da im simulierten OG ohne i3 mehr freie Zellen vorhanden sind und die Wahrscheinlichkeit einer Übereinstimmung der freien Zellen mit dem realen OG steigt.

Wider Erwarten steigen jedoch auch die 1-OE Ratio- und MS Ratio-Metrik. Für die 1-OE Ratio-Metrik sind zwei Gründe verantwortlich. Erstens wird die Metrik mit der Anzahl der belegten Zellen normiert, die in der Simulation ohne den BMW i3 geringer ist und somit zu besseren Ergebnissen führt. Der zweite Grund ist, dass der nicht normierte Wert OE ohne Berücksichtigung des i3 besser ist (OE: 567 ohne i3 statt 607 mit i3). Dies ist auch für die MS Ratio-Metrik gültig (MS: 22954 ohne i3 statt 22521 mit i3). Die Ursache hierfür ist, dass die beiden Metriken auf dem zellweisen Vergleich

der OGs basieren: Bei Berücksichtigung des i3 werden die Zellen des realen OGs mit dem Zellwert Eins verglichen, während die gleichen Zellen ohne Berücksichtigung des i3 mit dem Zellwert Null verglichen werden. Die durchschnittliche Abweichung zu den belegten Zellen ist also größer als die durchschnittliche Abweichung zu freien Zellen. Hinzu kommt, dass nicht alle belegten Zellen der Simulation auch eine reale belegte Zelle treffen, was die Metriken bei Berücksichtigung des i3 weiter verschlechtert (OE erhöht, MS verringert). Die Auswertung der anderen Fahrzeugabstände des gleichen Szenarios bestätigt das Verhalten der Metriken bei manueller Manipulation der Punktwolke.

Die drei Metriken, die direkt auf die Punktwolken angewendet werden, zeigen ebenfalls eine Verschlechterung ohne Berücksichtigung des i3 in der Simulation (siehe Abbildung 6-20). Die Ordinateachse ist in diesem Diagramm logarithmisch aufgetragen, da sich die Schwerpunkt-Distanz und Punktwolke-zu-Punktwolke-Distanz (PC2PC-Distanz) um eine Größenordnung vergrößern. Die Hausdorff-Distanz wird berechnet, um vorzuführen, warum die Anwendung dieser Metrik weiterhin nicht empfohlen wird. Die Änderung der Hausdorff-Distanz ist gering gegenüber den Änderungen der anderen Distanzen und wird durch Ausreißer bestimmt.

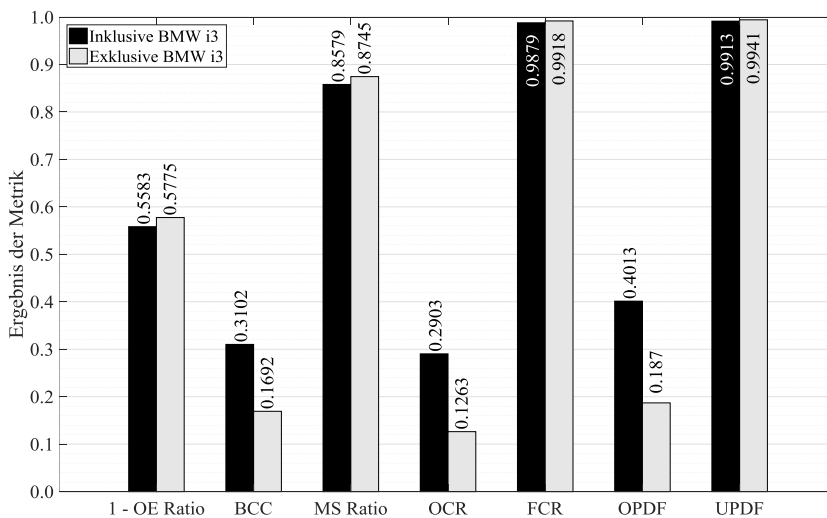


Abbildung 6-19: OG-Metriken für den Vergleich mit und ohne BMW i3, Szenario „Stauende 2 Fahrzeuge“

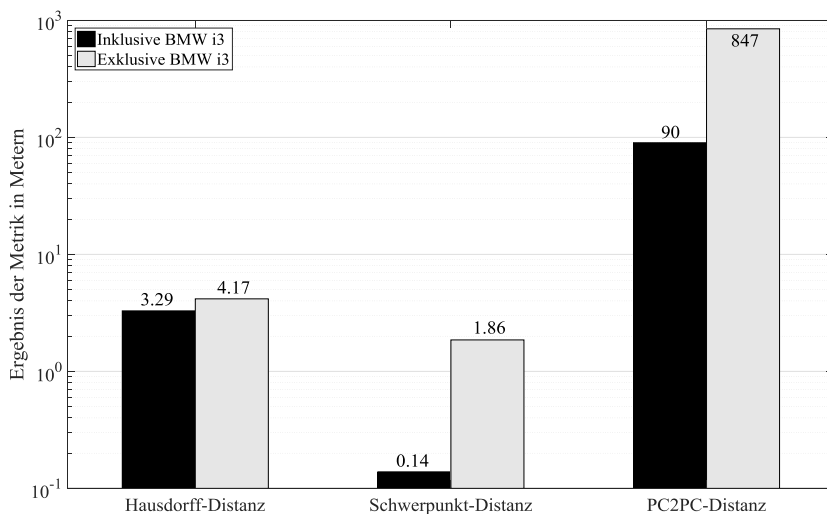


Abbildung 6-20: Punktwolken-Metriken für den Vergleich mit und ohne BMW i3, Szenario „Stauende 2 Fahrzeuge“

6.4.2 Tracking

Im Folgenden werden die in Abschnitt 5.4 vorgestellten Metriken zur Evaluation der Leistungsfähigkeit des implementierten Trackingalgorithmus von Knerr (im Folgenden „FZD-Tracker“ genannt) in Kombination mit den simulierten Punktwolken berechnet. Die Berechnung erfolgt anhand der in Abschnitt 6.3 definierten dynamischen Szenarien. Diese Szenarien werden in VTD simuliert und die simulierte Punktwolke als Eingang des implementierten (simulierten) FZD-Trackers verwendet. Die Ausgangsdaten des Trackers werden mit den Ausgangsdaten der realen Sensoren („ibeo-Tracker“) verglichen. Auf diese Weise werden die Tracking-Ergebnisse auf Basis der gesamten Verarbeitungsketten verglichen (siehe Abbildung 6-21). Der Einfluss der vorgelagerten Verarbeitungsebene „Punktwolken“ wird in dieser Betrachtung nicht beachtet, da nur die Objektlisten als Ausgang der Tracker verglichen werden. Es sei angemerkt, dass auch ein direkter Vergleich der Tracker (ibeo- und FZD-) mit den gleichen Eingangsdaten (bei beiden Trackern die reale Punktwolke) möglich ist, um explizit die Leistung des implementierten Trackingalgorithmus zu bewerten. Dies ist jedoch nicht das Ziel dieser Masterthesis und wurde bereits von Knerr¹²⁰ durchgeführt.

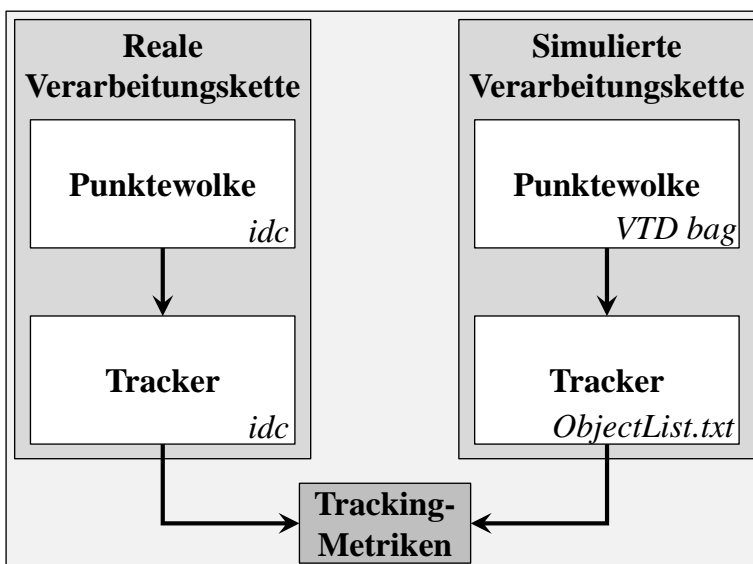


Abbildung 6-21: Methodik zum Vergleich der Tracking-Ergebnisse auf Basis der gesamten Verarbeitungsketten

Das erste Szenario, für welches die Tracking-Metriken berechnet werden, ist das Szenario „Stauende dynamisch“. Die manuell zeitlich korrelierten realen und simulierten Trajektorien der drei Fahrzeuge sind in Abbildung 6-22 dargestellt. Damit die Zeitschritte vergleichbar sind, werden alle Trajektorien einheitlich in 10 ms-Zeitschritten interpoliert. Außerdem werden die realen Daten für die Bewertung entsprechend Abschnitt 6.1.1 um die Z-Achse gedreht (Korrektur der Winkelabweichung). Ein zunächst qualitativer Vergleich der Trajektorien zeigt, dass die realen (ibeo-) Trajektorien eine bessere Trackkontinuität aufweisen und die Objekte vom ibeo-Tracker in größerer Entfernung erkannt werden.

¹²⁰ Vgl. Knerr, J.: Masterthesis, Entwicklung eines LiDAR-Tracking-Algorithmus für Sensormodelle (2018), S. 40–49.

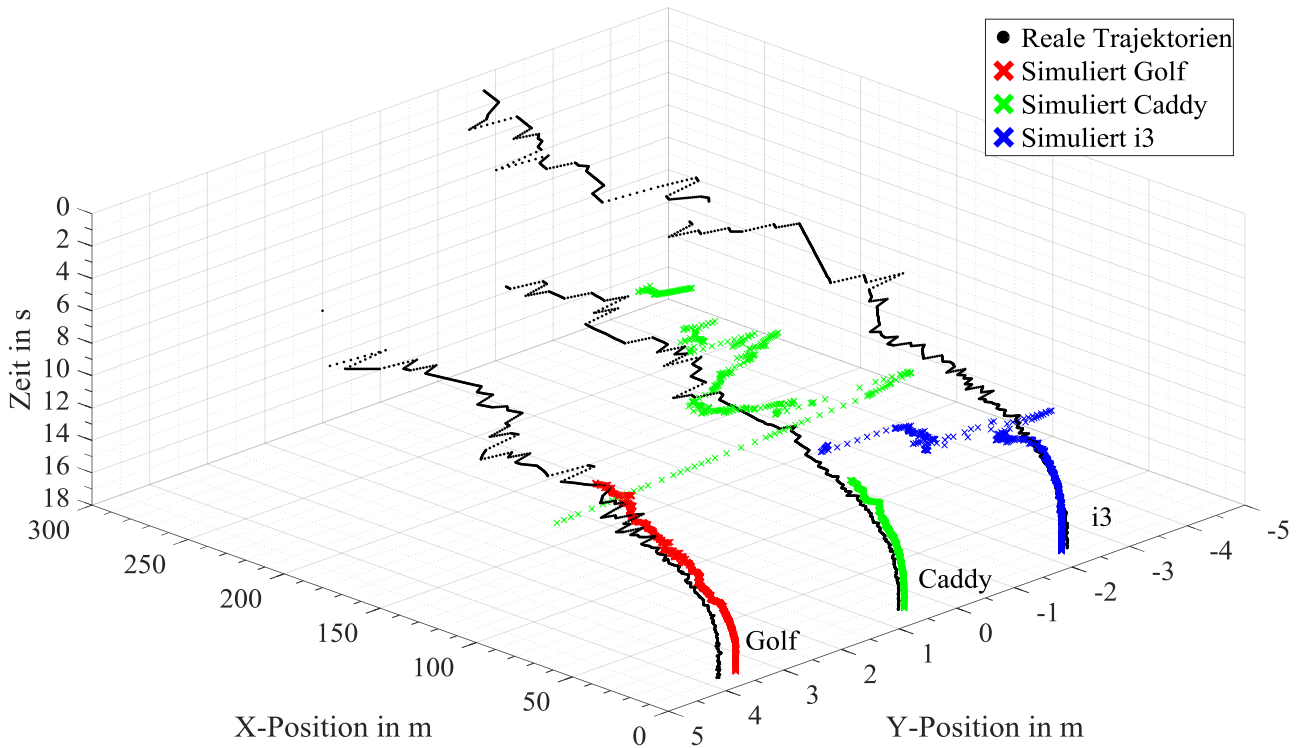


Abbildung 6-22: Reale und simulierte Trajektorien des Szenarios „Stauende dynamisch“

Für dieses Szenario werden die OSPA-, OSPA-T-, OSPA-MT-, GOSPA- und IoU-Metriken sowie der RMSE berechnet. Die Hausdorff- und OMAT-Metriken werden nicht berechnet, da die OSPA-Metrik auf diesen aufbaut und deren Schwachstellen eliminiert. Außerdem wird die Rahmathullah et al.-Metrik nicht berechnet, da diese eine hohe Ähnlichkeit mit der ebenfalls von Rahmathullah et al. definierten GOSPA-Metrik aufweist. Die Berechnung der Metriken erfolgt erst ab dem Zeitschritt, ab dem beide Tracker mindestens ein Objekt erkennen. Der ibeo-Tracker initiiert bereits ab circa 287 m eine Trajektorie, während der FZD-Tracker das erste Objekt erst 4,9 s später ab circa 160 m Entfernung erkennt.

Wie in Abschnitt 5.4.3 beschrieben wird die OSPA-Metrik mithilfe der Ungarischen Methode berechnet. Für jeden 10 ms-Zeitschritt existiert anschließend ein OSPA-Wert. Die auf der OSPA-Metrik basierenden Metriken OSPA-T und OSPA-MT werden ebenso ermittelt. Im unteren Diagramm der Abbildung 6-23 sind die drei Metriken in Abhängigkeit des Abstandes des mittleren Fahrzeugs (Caddy) vom Ego-Fahrzeug aufgetragen. Zusätzlich stellt das obere Diagramm in der Abbildung die vom Tracker des realen Sensors und vom Tracker des Sensormodells geschätzten Trajektorien dar. Das obere Diagramm in Abbildung 6-23 zeigt also den für die Metrikberechnung relevanten Ausschnitt des Diagramms in Abbildung 6-22, wobei bei ersterem auf die Zeitdimension verzichtet wird.

Für die Parametrierung der OSPA-Metriken wird $c = 3$ m, $p = 1$, $\Delta = 1$ m und $\alpha = 3$ m gewählt. Die Cut-Off-Distanz c wird auf drei Meter gesetzt, da dies ungefähr dem lateralen Abstand der Trajektorien zueinander entspricht. Für die Ordnung p wird eins gewählt, da dies direkt Rückschlüsse auf den euklidischen Abstand zwischen den realen und simulierten Trajektorien ermöglicht und Ausreißer nicht zu stark gewichtet werden. Der Parameter Δ gewichtet den Fehler, wenn eine reale Trajektorie

existiert und dieser keine simulierte Trajektorie zugewiesen wird. Dieser Parameter wird nach dem Vorschlag von Vu et al.¹²¹ gering gewählt, da somit der Lokalisierungsfehler einen größeren Einfluss besitzt. Vu et al. betonen jedoch, dass die Parametrisierung vom Szenario abhängig ist und bisher wenige Erfahrungswerte für die Parameter vorliegen. Der Parameter α der OSPA-T-Metrik ist für die Gewichtung des Benennungsfehlers verantwortlich und wird auf den maximalen Wert $\alpha = c$ gesetzt.

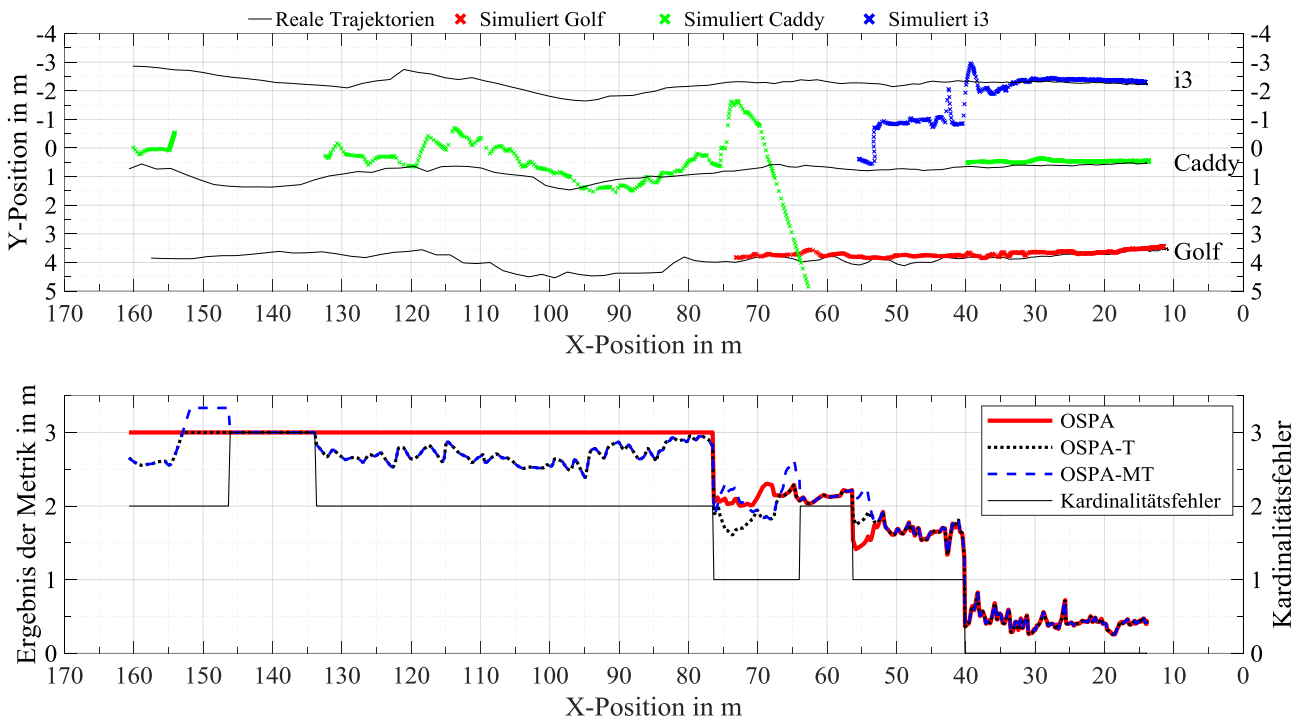


Abbildung 6-23: Trajektorien und OSPA-Metriken für das Szenario „Stauende dynamisch“

Der Vergleich der drei Metriken zeigt, dass alle vom Kardinalitätsfehler (dünne schwarze durchgezogene Linie in Abbildung 6-23 unten) bestimmt werden. Unterhalb von circa 40 m Entfernung der statischen Fahrzeuge zum Ego-Fahrzeug erkennen beide Tracker jeweils drei Trajektorien, somit ist der Fehler unterhalb dieses Abstands deutlich geringer. Die OSPA-Metrik kann grundsätzlich jeder realen Trajektorie maximal eine simulierte Trajektorie zuweisen. Deswegen sind die zwei weiter entfernten Teiltrajektorien des Caddy nicht berücksichtigt und die OSPA-Metrik nimmt oberhalb von circa 75 m Entfernung den maximalen Wert 3 m an (pro falscher Kardinalität wird 1 m addiert). Die OSPA-T- und OSPA-MT-Metriken hingegen berücksichtigen mehrere simulierte Trajektorien pro realer Trajektorie und sind daher weitestgehend identisch. Speziell in den Bereichen zwischen circa 75 m und 70 m sowie zwischen circa 57 m und 53 m Abstand des Ego-Fahrzeuges zu den statischen Fahrzeugen weichen die OSPA-T- und OSPA-MT-Distanzen jedoch voneinander ab. Die OSPA-MT-Metrik zeigt größere Werte, was intuitiv zu erwarten ist, da in diesen Bereichen die simulierte Trajektorie des Caddy bzw. des i3 deutlich von den realen Trajektorien abweicht. Die OSPA-T-Distanz hingegen ist geringer, da die OSPA-T-Metrik irrtümlich die falschen Trajektorien zuweist. Im ersten Bereich wird die simulierte Trajektorie des Caddy der realen Trajektorie des i3

¹²¹ Vgl. Vu, T. A.; Evans, R.: Optimal Subpattern Assignment Metric for Multiple Tracks (2018), S. 11.

zugewiesen und im zweiten Bereich umgekehrt, wodurch sich die Distanzen verringern. Die Trajektorien werden falsch zugewiesen, da der euklidische Abstand zur „falschen“ Trajektorie geringer ist. Dies verdeutlicht den Vorteil der OSPA-MT-Metrik, die nicht nur die einzelnen Abstände berechnet, sondern die gesamten Trajektorien berücksichtigt. Eine weitere Abweichung der OSPA-MT- zur OSPA-T-Distanz tritt im Bereich von circa 150 m Entfernung auf und ist auf einen größeren zulässigen Abstand zwischen realer und simulierter Trajektorie bei der OSPA-MT-Metrik zurückzuführen.

Die GOSPA-Metrik zeigt ein identisches Verhalten wie die OSPA-Metrik, entspricht jedoch mit den gewählten Parametern (für die GOSPA-Metrik wird nach der Empfehlung von Rahmathullah et al.¹²² $\alpha = 2$ gewählt) zu jedem Zeitschritt dem dreifachen Wert der OSPA-Distanz. Auch mit anderen Werten für α entspricht die GOSPA-Distanz der dreifachen OSPA-Distanz. Der Vorteil der GOSPA-Metrik gegenüber der OSPA-Metrik, dass eine Strafe für falsche und verfehlte Trajektorien berücksichtigt wird, ist in diesem Beispiel nicht ersichtlich, da keine falschen oder verfehlten Trajektorien vorhanden sind. Die GOSPA-Metrik wird in den folgenden Berechnungen nicht mehr berücksichtigt.

Der RMSE nach Morton et al.¹²³ wird für alle drei Trajektorien einzeln berechnet. Zuvor werden die zusammengehörigen realen und simulierten Trajektorien gelabelt. Die Trajektorie des Golf erzielt den geringsten RMSE, während die Trajektorie des Caddy den höchsten RMSE aufweist (siehe Tabelle 6-6). Der qualitative visuelle Vergleich der Ergebnisse mit den Trajektorien im oberen Diagramm der Abbildung 6-23 ist plausibel. Für eine erste Abschätzung der Abweichung der Trajektorien ist die Berechnung des RMSE nicht ungeeignet, die Trackkontinuität wird jedoch nicht beachtet. Außerdem erläutern Morton et al. die Vorgehensweise zur Berechnung des RMSE nicht, wenn bei einem Zeitschritt mehrere simulierte Trajektorien einer realen Trajektorie zugewiesen sind. Daher lässt sich der RMSE nur anwenden, wenn zu jedem Zeitschritt pro realer Trajektorie nur eine simulierte Trajektorie existiert.

Tabelle 6-6: RMSE für das Szenario „Stauende dynamisch“

	Trajektorie		
	VW Golf	VW Caddy	BMW i3
RMSE in m	0,582	1,195	0,775

Die letzte berechnete Metrik für die Bewertung des Trackingalgorithmus ist die Intersection-over-Union-Metrik (IoU-Metrik). Für jede Trajektorie wird für jeden Zeitschritt die Überdeckung der Bounding Boxes der Objekte in X- und Y-Richtung berechnet. Bei der Berechnung wird eine eventuell vorhandene Drehung der Bounding Box um die Z-Achse berücksichtigt. Die Ausdehnung der Box in Z-Richtung wird nicht berücksichtigt, da der ibeo-Tracker diese nicht ermittelt. Das obere Diagramm in Abbildung 6-24 zeigt die IoU-Metrik für die drei Trajektorien in Abhängigkeit des Abstandes des Ego-Fahrzeugs vom VW Caddy. Wenn keine Überdeckung der Bounding Boxes vorhanden ist, dann ist der IoU-Wert unabhängig vom Abstand der Bounding Boxes zueinander nicht

¹²² Vgl. Rahmathullah, A. S. et al.: Generalized optimal sub-pattern assignment metric (2017), S. 2.

¹²³ Vgl. Morton, P. et al.: Evaluation of dynamic object tracking (2011), S. 7.

definiert. Im unteren Diagramm der Abbildung 6-24 sind die Box-Plots der IoU-Werte der drei Trajektorien dargestellt. Die Trajektorie des VW Golf erzielt zwar den besten RMSE, der Median der IoU-Werte ist jedoch am geringsten, was auf große Unterschiede hinsichtlich der Größe der Bounding Boxes hinweist. In Verbindung mit dem RMSE ist also eine Aussage über die Güte der Schätzung der Größe der Bounding Box möglich. Es ist jedoch zu beachten, dass die Fahrzeugmodelle in der Simulation teilweise nicht realen Fahrzeugen entsprechen (siehe Fahrzeugäquivalente in Tabelle 6-4).

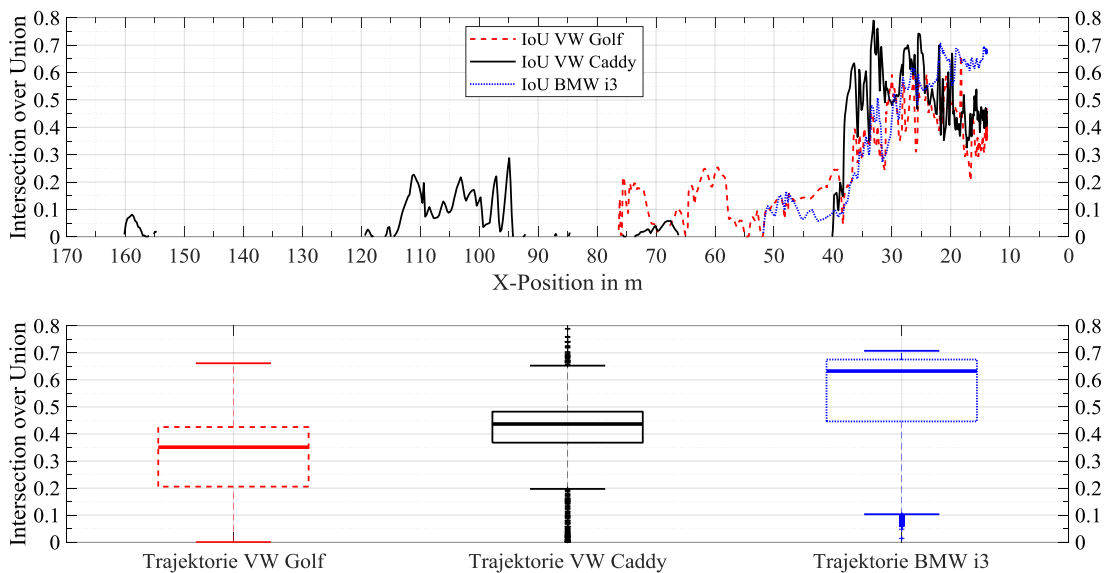


Abbildung 6-24: Intersection over Union in Abhängigkeit des Fahrzeugabstandes und Boxplots

Auch beim zweiten Szenario („i3 Schrittgeschwindigkeit“) zur Bewertung des Trackingalgorithmus werden die Metriken erst ab dem Zeitschritt berechnet, ab dem beide Tracker das Fahrzeug detektieren. Die Bewertung beginnt bei einer Distanz von circa 70 m, obwohl der ibeo-Tracker das Fahrzeug bereits ab circa 260 m durchgehend erkennt. Auch bei diesem Szenario werden die realen Daten gemäß Abschnitt 6.1.1 um die Z-Achse gedreht. Die Parametrisierung der OSPA-Metriken bleibt unverändert zum ersten Szenario. In Abbildung 6-25 sind die Trajektorien sowie die OSPA-Distanzen für dieses Szenario dargestellt. Da keine Unterbrechung der Trackkontinuität und keine falsche Zuweisung der Trajektorien vorliegen, sind die drei Metriken bei jedem Zeitschritt identisch. Alle Metriken werden mit abnehmendem Abstand des BMW i3 zum Ego-Fahrzeug tendenziell geringer, da sich die absolute Abweichung (euklidischer Abstand) der Trajektorien ebenfalls verringert. Bei der absoluten Abweichung hat die X-Komponente einen größeren Einfluss auf das Ergebnis als die Y-Komponente. Beide Komponenten verringern sich bei geringem Abstand zwischen i3 und Ego-Fahrzeug, wodurch die OSPA-, OSPA-T- und OSPA-MT-Distanzen insgesamt geringer ausfallen.

Der RMSE beträgt bei diesem Szenario 1,025 m und wird wie die OSPA-, OSPA-T- und OSPA-MT-Metriken zum Großteil vom Versatz der X-Komponente der Objektposition bestimmt. Dieser Versatz kommt zustande, da der FZD-Tracker die Fahrzeuglänge bei den meisten Zeitschritten größer einschätzt als der ibeo-Tracker und der Objektmittelpunkt daher weiter entfernt erscheint. Die Boxplots in den unteren Diagrammen der Abbildung 6-26 zeigen neben der Schätzung der Fahrzeuglänge auch die Schätzung der Fahrzeugbreite. Bei dieser liegt der Median der Schätzung des ibeo-Trackers nah

an der Ground Truth Fahrzeugbreite, während die Schätzung des FZD-Trackers deutlich geringer ausfällt und eine größere Schwankungsbreite aufweist.

Ebenfalls in Abbildung 6-26 aufgetragen ist die IoU-Metrik für das Szenario „i3 Schrittgeschwindigkeit“. Obwohl beide Trajektorien bereits ab circa 70 m Entfernung initiiert werden, besteht die erste Überdeckung der Bounding Boxes erst ab circa 50 m Entfernung. Außerdem ist die IoU-Metrik im Mittel geringer als beim ersten Szenario „Stauende dynamisch“.

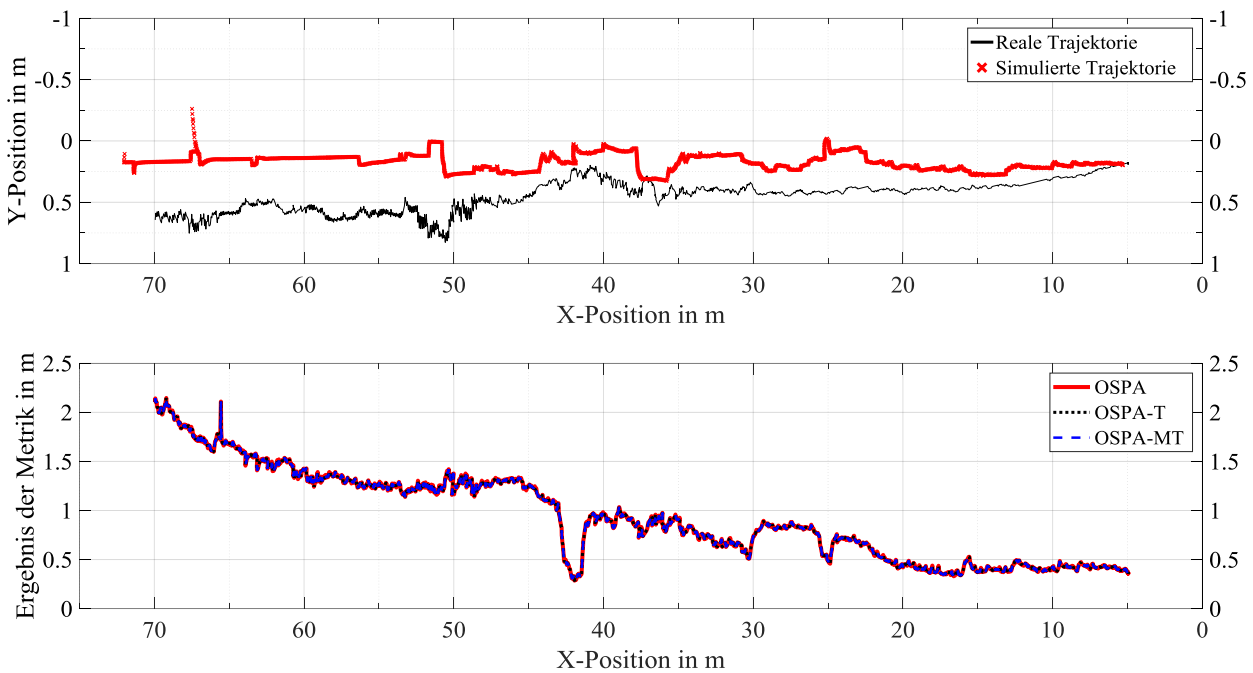


Abbildung 6-25: Trajektorien und OSPA-Metriken des Szenarios „i3 Schrittgeschwindigkeit“

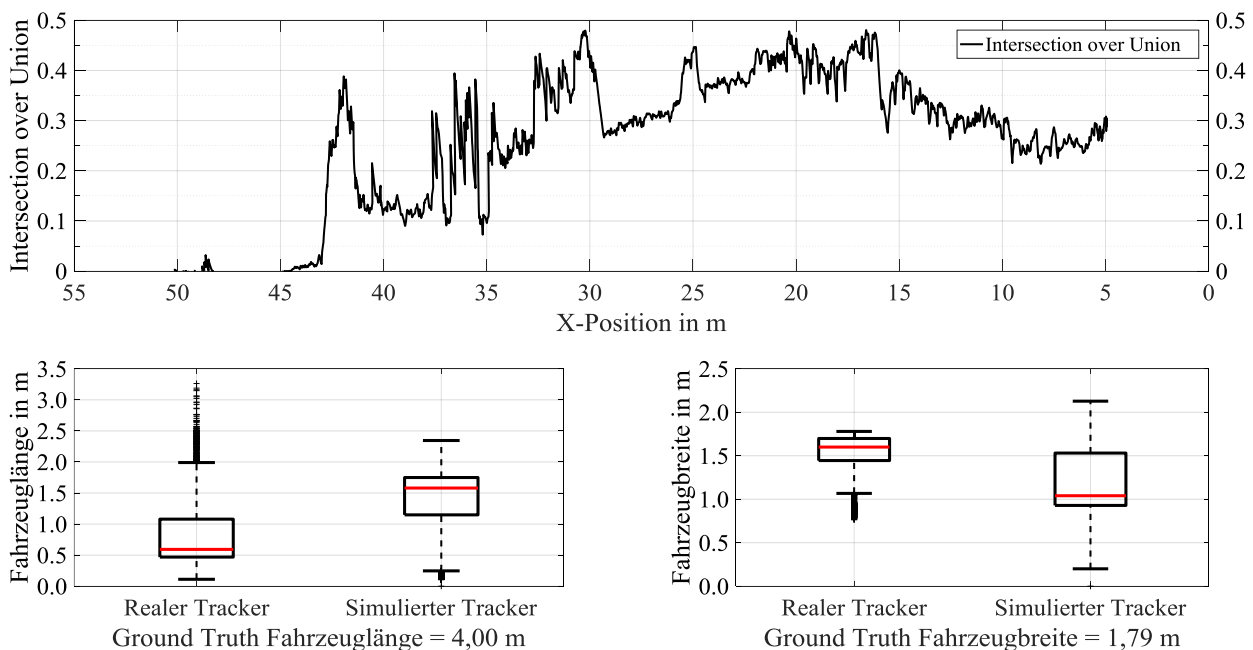


Abbildung 6-26: IoU-Metrik und Boxplots der Fahrzeuglänge und -breite des Szenarios „i3 Schrittgeschwindigkeit“

Zusammenfassend wird festgehalten, dass bei der Bewertung des Trackingalgorithmus eine Berücksichtigung mehrerer Metriken empfohlen wird. Falls nur eine simulierte Trajektorie pro realer Trajektorie vorhanden ist, wird für eine einfache Abschätzung der RMSE berechnet, da bei diesem nur ein Wert zu interpretieren ist. Der IoU-Wert ermöglicht eine Abschätzung der Überdeckung der Bounding Boxes, wobei eine gleichzeitige Auswertung der geschätzten Fahrzeuglänge und -breite, z. B. in Form von Boxplots wie in Abbildung 6-26 gezeigt, empfohlen wird. Außerdem existiert bei der IoU-Metrik pro Zeitschritt ein Wert, der eine Interpretation erfordert. Mithilfe von Boxplots lässt sich die Interpretation auf wenige Werte vereinfachen (siehe Abbildung 6-24).

Speziell bei der Betrachtung der Bounding Boxes ist zu berücksichtigen, dass die Initialisierung eines Objekts mit festgelegten Startwerten für die Fahrzeuglänge und -breite erfolgt. Diese Startwerte betragen bei dem FZD-Tracker 0 m, während die Startwerte des ibeo-Trackers nicht bekannt sind. Hierdurch entstehen Abweichungen zwischen den Bounding Boxes, die bei neu initialisierten Objekten deutlich werden. Unterschiedlich initialisierte Fahrzeuglängen und -breiten führen auch zu abweichenden Referenzpunkten der Objekte, die sich auf die gesamte Trajektorie und somit auf die Ergebnisse aller Trackingmetriken auswirken können.

Wenn die Trajektorien Unterbrechungen aufweisen (siehe Szenario „Stauende dynamisch“) oder gleichzeitig mehrere simulierte Trajektorien existieren, die einer realen Trajektorie zugewiesen sind, dann wird die Verwendung der OSPA-Metriken empfohlen. Speziell die OSPA-MT-Metrik wird aufgrund der Berücksichtigung mehrerer gleichzeitig vorhandener Trajektorien ohne falsche Zuweisung der Trajektorien bevorzugt. Vor der Anwendung der OSPA-Metriken ist eine Definition der Parameter, insbesondere c und p , erforderlich, da deren Einfluss auf das Ergebnis nicht zu vernachlässigen ist.

6.4.3 Kombinierte Betrachtung der Verarbeitungsebenen „Punktwolken“ und „Tracking“

Für die kombinierte Betrachtung der Verarbeitungsebenen „Punktwolken“ und „Trackingalgorithmus“ werden sowohl die simulierten als auch die realen Punktwolken als Eingang des FZD-Trackers verwendet (siehe Abbildung 6-27). Hierdurch wird überprüft, wie sich die erste Verarbeitungsebene der Lidar-Sensorsimulation (Punktwolken) auf die Verarbeitungsebene Tracking auswirkt und ob Korrelation zwischen den Metriken der beiden Verarbeitungsebenen vorhanden ist. Zunächst werden die Ergebnisse des Trackings analog zum vorherigen Abschnitt 6.4.2 analysiert. Daraufhin werden an allen Zeitpunkten des Szenarios die Punktwolken betrachtet und die in Abschnitt 6.4.1 gezeigten Metriken angewendet. Im Idealfall sind die zwei Objektlisten, die der FZD-Tracker ausgibt, identisch und somit die Tracking-Metriken ideal (OSPA-Distanzen = 0 m und IoU = 1). Dies würde bedeuten, dass die Eingangsdaten sehr ähnlich sind und somit die Lidar-Sensorsimulation ein ähnliches Abbild des realen Sensors ist. Abweichungen, die sich in den Objektlisten ergeben, sind also auf unterschiedliche Eingangsdaten zurückzuführen. Somit kann überprüft werden, wie sich die erste Verarbeitungsebene auf das Tracking auswirkt. Die vorgestellte Methode wird beispielhaft am Szenario „Stauende dynamisch“ berechnet.

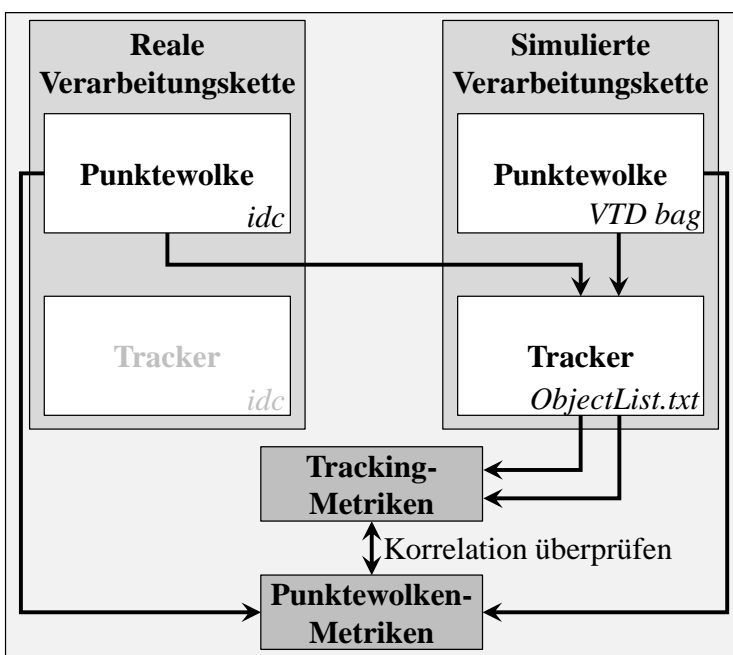


Abbildung 6-27: Methodik zur kombinierten Betrachtung der Verarbeitungsebenen

Das obere Diagramm der Abbildung 6-28 zeigt die Trajektorien, die mit realen und simulierten Punktwolken als Eingangsdaten des FZD-Trackers für das Szenario „Stauende dynamisch“ entstehen. Die Trajektorie des Caddy mit realen Eingangsdaten wird bereits in circa 220 m Entfernung initialisiert, für den Vergleich mit den OSPA-Metriken jedoch auf circa 160 m gekürzt. Der qualitative Vergleich zeigt, dass die Trajektorien mit realen Eingangsdaten eine bessere Trackkontinuität aufweisen. Außerdem werden zwei von drei Trajektorien (Golf und Caddy) mit realen Eingangsdaten in größerer Entfernung initialisiert.

Im mittleren Diagramm der Abbildung 6-28 sind die OSPA-, OSPA-T- und OSPA-MT-Distanzen dargestellt. Die Parameter der OSPA-Metriken entsprechen denen der Tracking-Bewertung in Abschnitt 6.4.2. Die Metriken zeigen ein ähnliches Verhalten wie beim Vergleich der gesamten Verarbeitungsketten (siehe hierzu Abbildung 6-23). Auch hier ist die falsche Zuweisung der OSPA-T-Metrik in den Bereichen 69 – 64 m und 57 – 43 m erkennbar. Die OSPA-MT-Distanz zeigt häufig dann Sprünge, wenn sich der Kardinalitätsfehler aufgrund einer neu initialisierten Trajektorie verringert (z.B. beim Abstand 77 m oder 41 m).

Im unteren Diagramm der Abbildung 6-28 sind die IoU-Werte für die drei Trajektorien dargestellt. Auch diese verbessern sich unterhalb eines Abstands von circa 40 m, ab dem alle drei Objekte vom FZD-Tracker mit realen oder simulierten Eingangsdaten erfasst werden. Der IoU des BMW i3 sinkt vom Maximalwert von circa 0,9 bei 20 m Distanz auf circa 0,5 bei 14 m Distanz, obwohl sich der euklidische Abstand der Objektmittelpunkte nicht deutlich vergrößert. Dies wird mithilfe der Metriken für Punktwolken überprüft.

Vor der Berechnung der Punktwolken-Metriken werden die realen und simulierten Punktwolken miteinander zeitlich korreliert. Anschließend werden die Punktwolken-Metriken mit den Tracking-Metriken zeitlich korreliert. Die erste Korrelation erfolgt unter Verwendung der Simulationsausgabe „SimTime“ (Zeitschritte sind nicht identisch mit den „ROSTime“-Zeitschritten!). Da die Aufzeichnungsfrequenzen der Punktwolken nicht gleich sind (reale Punktwolke circa 12,5 Hz, simulierte Punktwolke circa 25 Hz), wird jeder realen Punktwolke die zeitlich nächstgelegene simulierte Punktwolke zugeordnet. Für alle Punktwolkenpaare werden die Schwerpunkt- sowie Punktwolke-zu-Punktwolke-Distanzen (PC2PC-Distanz) ermittelt und mit den aus den Punktwolken generierten Scan Grids die 1-OE Ratio-, BCC-, OCR-, FCR-, OPDF3- und UPDF3-Metriken berechnet. Die Punktwolken und Scan Grids (Zellgröße 0,1 m) werden hierbei auf den relevanten Bereich reduziert, der die drei statischen Fahrzeuge sowie einen Randbereich um die Fahrzeuge umfasst.

Die Korrelation der Punktwolken-Metriken mit den Tracking-Metriken erfolgt über die Zeitstempel der realen Punktwolke und der Objektliste des FZD-Trackers mit der realen Punktwolke als Eingang. In beiden ist die Zeit für jeden Punkt bzw. jede ID gespeichert, wobei die Zeitrechnung der Punktwolke am 01.01.1900 00:00:00 UTC beginnt, während die Zeitrechnung der Objektliste mit der Unixzeit (Startzeitpunkt am 01.01.1970 00:00:00 UTC) übereinstimmt. Die Differenz dieser Startzeitpunkte beträgt unter Berücksichtigung von 17 Schaltjahren 2208988800 s und wird von den Zeitstempeln der realen Punktwolke subtrahiert. Anschließend wird für jeden Zeitpunkt, bei dem die Tracking-Metriken berechnet sind, der nächstgelegene Zeitpunkt ermittelt, bei dem die Punktwolken-Metriken berechnet sind.

In Abbildung 6-29 sind die genannten Punktwolken-Metriken in Abhängigkeit des Abstandes der statischen Fahrzeuge zum Ego-Fahrzeug dargestellt. Insgesamt verbessern sich die Metriken mit sinkendem Abstand der Fahrzeuge zueinander. Ab circa 20 m Entfernung verschlechtern sie sich jedoch wieder, was sich auch bei den Tracking-Metriken (hauptsächlich IoU und geringe Vergrößerung der OSPA-MT-Distanz) in Abbildung 6-28 bemerkbar macht. Die Punktwolke-zu-Punktwolke-Distanz ist in diesem Beispiel mit der maximalen Punkteanzahl der realen oder simulierten Punktwolke normiert, da sonst aufgrund der steigenden Punkteanzahl mit sinkender Entfernung eine Auswertung

schwierig ist (Abstand 160 m: < 10 Punkte, Abstand 15 m: > 400 Punkte). Die FCR- und UPDF3-Metriken sind während des gesamten Szenarios nahezu 1,0 und verschlechtern sich geringfügig mit abnehmendem Abstand, da insgesamt weniger freie Zellen vorhanden sind.

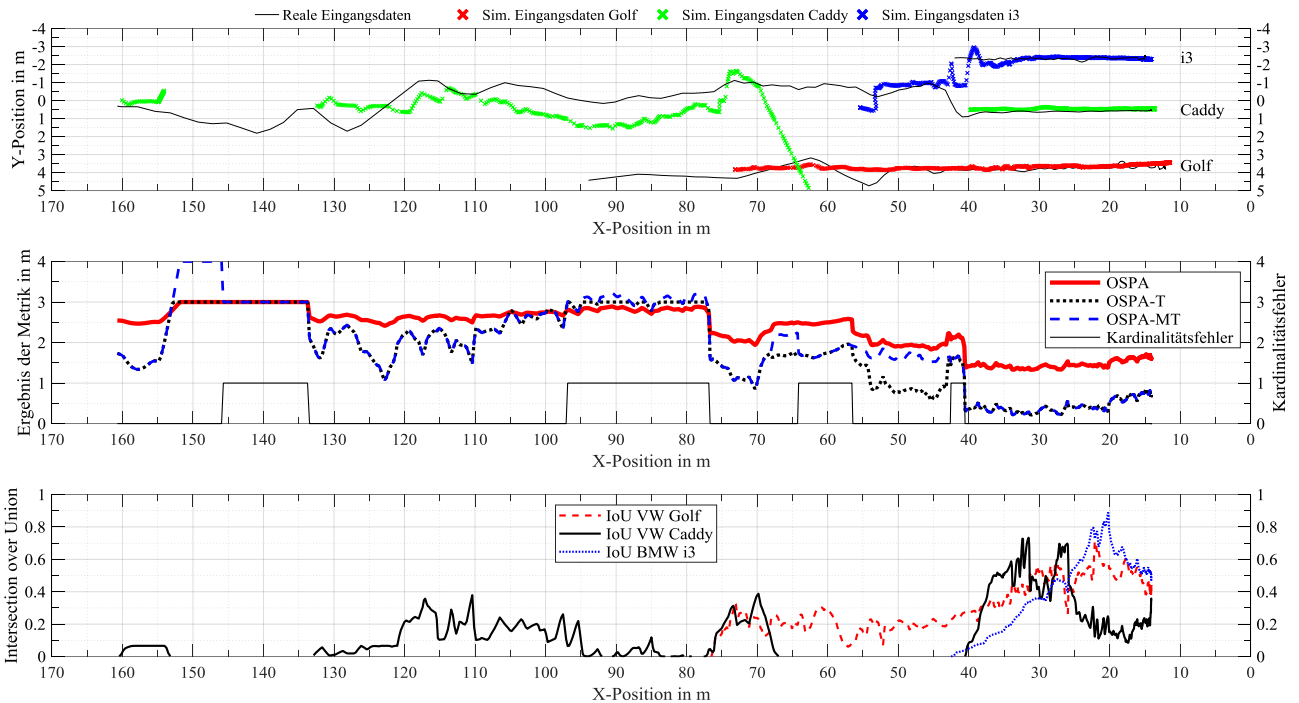


Abbildung 6-28: Tracking-Metriken für das Szenario „Stauende dynamisch“, FZD-Tracker mit realen und simulierten Punktwolken als Eingangsdaten

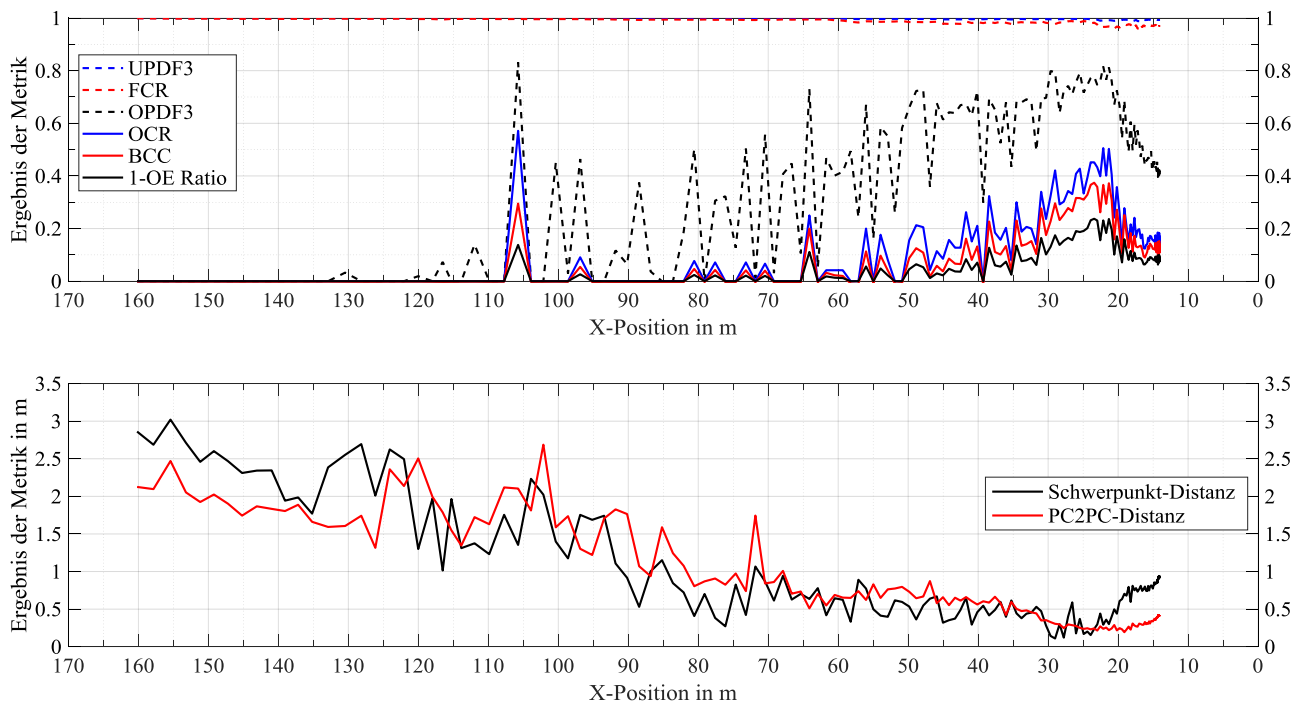


Abbildung 6-29: Punktwolken-Metriken für das Szenario „Stauende dynamisch“

Abschließend wird die Korrelation zwischen den Punktwolken- und Tracking-Metriken untersucht. Dies erfolgt mittels des Korrelationskoeffizienten nach Pearson, der in Tabelle 6-7 für verschiedene Kombinationen von Punktwolken- und Tracking-Metriken dargestellt ist. Negative Korrelation bedeutet, dass der Wert einer Metrik größer wird, während der Wert der anderen Metrik kleiner wird. Dies ist bei bestimmten Kombination erwünscht, da bei der OSPA-MT-, Schwerpunkt- und PC2PC-Distanz geringere Werte besser sind (minimal 0,0 m), während bei der IoU-, 1-OE Ratio-, BCC-, OCR- und OPDF3-Metrik größere Werte besser sind (maximal 1,0). Bei Berechnung des Korrelationskoeffizienten mit dem IoU sind die IoU-Werte der drei Trajektorien von Golf, Caddy und i3 gemittelt.

Der Korrelationskoeffizient zeigt bei allen Kombinationen eine hohe Korrelation, wobei die Korrelation der Punktwolken-Metriken mit dem IoU tendenziell besser ist als mit der OSPA-MT-Distanz. Insgesamt deutet der Korrelationskoeffizient darauf hin, dass der Einfluss der Qualität der Punktwolken auf das Tracking-Ergebnis nicht zu vernachlässigen ist. Eine geringe Übereinstimmung der realen und der simulierten Punktwolke führt demnach auch zu einer größeren Abweichung der Trajektorien beim Tracking. Es wird auch bei der kombinierten Betrachtung empfohlen, mehrere Metriken parallel zu betrachten.

Tabelle 6-7: Korrelationskoeffizienten zwischen den Punktwolken- und Tracking-Metriken

Korrelationskoeffizienten		Punktwolken-Metriken					
		1-OE Ratio	BCC	OCR	OPDF3	Schwerpunkt-Distanz	PC2PC-Distanz
Tracking-Metriken	OSPA-MT	-0,717	-0,696	-0,666	-0,698	0,577	0,773
	IoU	0,797	0,768	0,725	0,692	-0,588	-0,780

7 Diskussion der Ergebnisse

In diesem Kapitel werden die Ergebnisse und die Erkenntnisse dieser Masterthesis kritisch diskutiert. Dabei werden zum einen die erzielten Ergebnisse bei der Bewertung und zum anderen die Fehler betrachtet, die bei der Durchführung der realen Fahrversuche und der Berechnung der Metriken auftreten können. Außerdem wird der Einfluss der Fehler auf die Ergebnisse diskutiert.

Der Vergleich der Sensorsimulation mit den realen Daten auf der Verarbeitungsebene der Punktwolken in Abschnitt 6.4.1 erfolgt hauptsächlich mit den aus den Punktwolken ermittelten Occupancy Grids. Für die Erstellung der Occupancy Grids werden zwei Varianten vorgestellt. Bei der ersten Variante wird für jeden Scan ein Scan Grid erstellt, welches anschließend bewertet wird. Da diese Variante nicht den Rauscheffekt der Punktwolke berücksichtigt, wird eine zweite Variante entwickelt, bei der alle Scan Grids gemittelt werden. Die zweite Methode lässt sich auch anwenden, wenn in zukünftigen Versionen des Lidar-Sensormodells Sensorrauschen implementiert wird. In diesem Fall wird jeweils aus den realen und simulierten Punktwolken eine zuvor definierte Anzahl von Scan Grids erstellt und anschließend diese jeweils gemittelt, damit ein einziges reales Occupancy Grid mit einem einzigen simulierten Occupancy Grid verglichen wird.

Bei der Erstellung der Scan Grids bzw. Occupancy Grids existieren mehrere Fehlerquellen. Der erste Fehler entsteht bei der Anzahl der Scans, die bei der Mittelung der Scan Grids berücksichtigt werden. Die Untersuchungen haben jedoch ergeben, dass sich die Ergebnisse der Metriken ab einer Anzahl von 100 Scan Grids nur geringfügig ändern. Der zweite Fehler entsteht bei der Festlegung der Zellauflösung. Die Variation der Zellauflösung zeigt, dass die Zellgröße einen großen Einfluss auf das Ergebnis der Metriken hat. Grundsätzlich wird jedoch die Zellgröße 0,04 m gewählt, da dies der Distanzauflösung des verbauten Lidar-Sensors entspricht. Ein dritter Fehler bei der Erstellung der Scan Grids entsteht, weil die Anzahl der Punkte, die sich in einer Zelle befinden, nicht beachtet wird. Somit bleibt bei der Berechnung der Metriken für die Scan Grids unberücksichtigt, ob sich in einer Zelle nur ein Punkt befindet oder beispielsweise hundert Punkte. Dieser Fehler ist bei der direkten Betrachtung der Punktwolken nicht vorhanden. Ein weiterer Fehler bei den Scan Grids ist, dass die Zellen nur mit binären Werten befüllt werden, wenn sich innerhalb einer Zelle ein Punkt befindet. Hierbei bleibt unberücksichtigt, ob die Zelle im Sichtbereich des Sensors liegt oder von Objekten verdeckt ist, die zwischen der Zelle und dem Sensor liegen. Dadurch werden alle Zellen hinter einer belegten Zelle als „frei“ angenommen, obwohl deren Status „unbekannt“ ist. Eine direkte Berücksichtigung der verdeckten Zellen bei der Erstellung der Scan Grids eliminiert diesen Fehler, erhöht jedoch den Rechenaufwand, da die Verdeckung bezüglich zweier Lidar-Sensoren berechnet werden muss.

Bei den Metriken, die direkt auf die Punktwolken angewendet werden und diese vergleichen, wirkt sich die Anzahl der Punkte teilweise auf das Ergebnis aus. Als Beispiel hierfür sei die Punktwolke-zu-Punktwolke-Distanz genannt, deren Wert mit der Anzahl der Punkte skaliert. Der Abstand der Schwerpunkte der Punktwolken kann genutzt werden, um grundsätzliche Fehler zwischen den Punktwolken zu detektieren. Zuvor muss sichergestellt werden, dass die Position des Schwerpunkts einer

Punktwolke nicht durch Bodenreflexionen beeinflusst wird. Die Hausdorff-Distanz ist sensitiv gegenüber Ausreißern und eine Verwendung für weitere Punktwolken-Vergleiche wird nicht empfohlen.

Auch bei der Berechnung der Tracking-Metriken in Abschnitt 6.4.2 unterlaufen Fehler, die in der folgenden Fehlerbetrachtung analysiert werden. Der erste Fehler entsteht bei der Interpolation der realen und simulierten Trajektorien auf die 10 ms-Zeitschritte. Außerdem entsteht ein Fehler bei der zeitlichen Korrelation der realen und simulierten Trajektorien. Dieser wird minimiert, indem mehrere Zeitschritte überprüft werden. Die Berechnung der OSPA-Metriken erfordert eine vorherige Definition der Berechnungsparameter. Da für die Parameter bisher wenige Erfahrungswerte vorliegen, werden in dieser Masterthesis die Parameter entsprechend den Empfehlungen der Autoren der Metriken gewählt. Die Wahl der Parameter beeinflusst die absoluten Werte der Metriken teilweise deutlich, wobei die relativen Verläufe beibehalten werden und somit der Vergleich verschiedener Simulationen und Szenarien möglich ist.

Die Berechnung des Intersection-over-Union-Wertes zeigt bei größeren Distanzen eine geringe Übereinstimmung der Bounding Boxes. Dies liegt neben der Abweichung der Objektmittelpunkte an unterschiedlich geschätzten Fahrzeuglängen und -breiten. Während die Startwerte der Länge und Breite beim FZD-Tracker bei 0 m beginnen, sind die Startwerte des ibeo-Trackers nicht bekannt.

Da auch bei der kombinierten Betrachtung der beiden Verarbeitungsebenen Punktwolken und Trackingalgorithmus die Daten korreliert werden, sind auch hier Fehler vorhanden. Die Berechnung der Metriken der realen und simulierten Punktwolken erfolgt an diskreten Zeitpunkten, die aufgrund der unterschiedlichen Aufzeichnungsfrequenz nicht immer übereinstimmen. Die maximale Differenz zwischen zwei Zeitpunkten, an denen jeweils ein realer und simulierter Scan aufgezeichnet werden, beträgt 0,0236 s. Diese Differenz tritt bei einem Abstand der Fahrzeuge von circa 117 m auf und das Ego-Fahrzeug fährt zu diesem Zeitpunkt mit circa 100 km/h. Die Strecke, die es bei dieser Geschwindigkeit in der Zeitdifferenz 0,0236 s zurücklegt, beträgt 0,657 m. Diese Distanz entspricht also der maximalen Abweichung zwischen der simulierten und realen Punktwolke durch den Fehler bei der Korrelation. Die durchschnittliche Zeitdifferenz beträgt circa 1/3 der maximalen Zeitdifferenz. Bei geringeren Abständen ist die Fahrzeuggeschwindigkeit geringer, sodass sich der Fehler hierdurch zusätzlich verkleinert.

Ein weiterer Fehler wirkt sich nicht auf die Berechnung der Metriken aus, sondern nur auf die anschließende Korrelation zwischen den Punktwolken- und Tracking-Metriken. Die maximale Zeitdifferenz zwischen zwei Zeitpunkten, für die die Punktwolken- und Tracking-Metriken berechnet sind, beträgt 0,001837 s bei einem Abstand von circa 18,5 m des Ego-Fahrzeugs zu den statischen Fahrzeugen. Bei dieser Distanz ist die Geschwindigkeit des Ego-Fahrzeugs auf circa 13,5 km/h verringert und die Abweichung beträgt demnach nur 0,0069 m. Die maximale Abweichung beträgt 0,038 m bei einem Abstand von circa 155 m zwischen Ego-Fahrzeug und statischen Fahrzeugen, da hier die Fahrzeuggeschwindigkeit noch größer ist. Diese Abweichung infolge der Korrelation zwischen den Punktwolken- und Tracking-Metriken ist in den Abbildungen aufgrund der geringen Werte nicht zu erkennen und auch der Einfluss dieser Abweichung auf die Berechnung des Korrelationskoeffizienten von Punktwolken- und Tracking-Metriken kann vernachlässigt werden.

Alle mit den realen Lidar-Sensoren aufgezeichneten Punktwolken und Trajektorien werden in dieser Masterthesis um $+1,7204^\circ$ um die Z-Achse gedreht. Dieser Winkel wird bei Messungen mit dem *Corner Reflector* ermittelt. Auch diese Messungen unterliegen einem Messfehler, z. B. durch die Ausrichtung des Versuchsfahrzeuges, die Positionierung des *Corner Reflectors* oder die anschließende Ermittlung des Schwerpunkts des *Corner Reflectors* in der Punktwolke. Der ermittelte Winkel ist also nur ein erster Versuch zur Quantifizierung der Winkelabweichung und entspricht nicht der wahren Winkelabweichung. Diese Winkelabweichung besitzt jedoch einen großen Einfluss auf die Metriken und daher besteht die Forderung, die Winkelabweichung bei jeder Interpretation der Ergebnisse zu berücksichtigen.

8 Fazit und Ausblick

In dieser Masterthesis wird eine Methodik entwickelt, die die Bewertung der Lidar-Sensorsimulation auf unterschiedlichen Verarbeitungsebenen ermöglicht. Zunächst werden die theoretischen Grundlagen des Lidar-Sensormodells bei FZD erläutert. Es wird die Anwendbarkeit einer FMEA für die Bewertung einer Sensorsimulation überprüft und die FMEA beispielhaft durchgeführt. Eine FMEA ist grundsätzlich durchführbar, jedoch nicht in vollem Umfang, da hierzu viele Informationen zur Sensorsimulation fehlen. Dennoch zeigt die FMEA die Zusammenhänge zwischen den Verarbeitungsebenen. Anschließend werden unterschiedliche Validierungsmethoden diskutiert und darauf aufbauend eine Methodik zur Bewertung der Lidar-Sensorsimulation entwickelt.

In einer ausführlichen Literaturrecherche werden Metriken zur Bewertung der Lidar-Sensorsimulation auf den vier Verarbeitungsebenen vorgestellt. Anschließend wird überprüft, ob aus den einzelnen Metriken ein Gesamtmaß entwickelt werden kann. Das Ergebnis dieser Untersuchung ist, dass kein Gesamtmaß erstellt wird, da bei einem Gesamtmaß die Informationen über die Schwachstellen und Fehler der einzelnen Verarbeitungsebenen verloren gehen und ein Gesamtmaß physikalisch schwierig interpretierbar ist.

Die vorgestellten Metriken für die unterschiedlichen Verarbeitungsebenen werden an einfachen beispielhaften Szenarien angewendet. Hierzu werden bei realen Versuchsfahrten Messdaten aufgezeichnet und diese Szenarien anschließend mit dem Simulationstool VTD nachgestellt.

Für die Realdaten wird die Winkelabweichung der Punktwolken in einem zusätzlichen Versuch bestimmt. Diese beträgt $1,7204^\circ$ und wird in allen Auswertungen berücksichtigt. Außerdem wird die Genauigkeit der Delta-Distanz zwischen zwei ADMA bestimmt, da diese in vorherigen Arbeiten größere Abweichungen gezeigt hat. In mehreren Versuchen wird gezeigt, dass die Genauigkeit für die Verwendung der ADMA-Daten bei der Bewertung der Lidar-Sensorsimulation ausreichend ist.

Die Anwendung der Metriken erfolgt nur auf den beiden Verarbeitungsebenen „Punktwolken“ und „Trackingalgorithmus“, da bei der Verarbeitungsebene „Segmentierung“ die Segmente nicht explizit als Ausgabe zur Verfügung stehen. Die Verarbeitungsebene „Klassifizierungsalgorithmus“ wird nicht bewertet, da der implementierte Algorithmus lediglich zwei Klassen unterscheidet und somit deutlich weniger Klassen erkennt als der Algorithmus der im Versuchsfahrzeug verbauten Lidar-Sensoren.

Die Metriken zur Bewertung der Verarbeitungsebene „Punktwolken“ zeigen, dass die Werte von vielen Faktoren wie z. B. der Zellgröße der Occupancy Grids abhängig sind. Außerdem ist für eine Vergleichbarkeit verschiedener Simulationen eine Normierung vieler Metriken erforderlich. Einzelne Metriken wie z. B. der Map Score lassen sich ohne eine Umformulierung nicht anwenden. Die Interpretation der Ergebnisse der Metriken ist schwierig, da noch keine Erfahrungswerte darüber vorhanden sind, welche Werte der Metriken eine gute Lidar-Sensorsimulation auszeichnen. Ein Vergleich der Szenarien untereinander ist jedoch möglich und außerdem können die Metriken für einen Vergleich unterschiedlicher Sensorsimulationen verwendet werden. Die Sensitivität der Metriken wird an einem einfachen Szenario überprüft, indem ein Fahrzeug aus der simulierten Punktwolke entfernt wird. Für eine weitere Verwendung werden die 1-OE Ratio-, BCC-, MS Ratio-, OCR-, FCR-, OPDF-

und UPDF-Metriken sowie die Schwerpunkt- und Punktwolke-zu-Punktwolke-Distanzen empfohlen. Eine parallele Betrachtung mehrerer Metriken stellt sich als vorteilhaft heraus, da sich die Metriken bei verschiedenen Szenarien unterschiedlich verhalten und eine einzige Metrik nicht alle möglichen Fälle abdeckt.

Die dynamischen Szenarien zur Bewertung der Verarbeitungsebene „Trackingalgorithmus“ zeigen die Unterschiede zwischen der OSPA-, OSPA-T und OSPA-MT-Metrik. Eine Verwendung der OSPA-MT-Metrik wird empfohlen, da diese mehrere simulierte Trajektorien pro realer Trajektorie in der Berechnung berücksichtigt. Die OSPA-Metriken erfordern eine Definition der Berechnungsparameter, da diese einen großen Einfluss auf das Ergebnis der Metriken besitzen. Aufgrund fehlender Erfahrungswerte wird für die Parameterwahl auf die Empfehlungen der Autoren der Metriken zurückgegriffen. Da in der Simulation ein *Extended Object Tracker* implementiert ist, der die Objektlänge und -breite schätzt, wird mithilfe der Intersection-over-Union-Metrik die Größe und Position der Bounding Box bewertet. Die Untersuchung zeigt, dass die Schätzungen der Fahrzeuglänge beim ibeo- und FZD-Tracker deutlich von der wahren Fahrzeuglänge abweichen, da die Fahrzeuge bei dem Szenario längs vor dem Versuchsfahrzeug ausgerichtet sind. Bei der Schätzung der Fahrzeugbreite liegt der Wert des ibeo-Trackers nahe an der wahren Fahrzeugbreite, während der FZD-Tracker deutlich davon abweicht.

In der letzten Betrachtung werden die Punktwolken- und Tracking-Metriken kombiniert ausgewertet. Anhand eines beispielhaften Szenarios werden für alle Zeitschritte die Punktwolken- und Tracking-Metriken berechnet und die Korrelation zwischen diesen ermittelt. Bei allen Metriken zeigt sich eine hohe Korrelation ($> 0,5$), die auf einen großen Einfluss der Qualität der Lidar-Sensorsimulation (simulierte Punktwolke) auf das Ergebnis des Trackingalgorithmus hindeutet.

Auf Basis dieser Masterthesis sind weitere Untersuchungen zur Bewertung einer Lidar-Sensorsimulation mit verschiedenen Verarbeitungsebenen möglich. Der folgende Ausblick zeigt, in welchen Bereichen noch Potential zur Weiterentwicklung der Methodik oder der Metriken vorhanden ist.

Während dieser Masterthesis wird eine Winkelabweichung der realen Punktwolke erkannt und in einem Versuch erstmalig quantifiziert. Es zeigt sich, dass trotz der Drehung der realen Punktwolken um den ermittelten Winkelfehler ($1,7204^\circ$) weiterhin eine Abweichung zur Ground Truth besteht. Vor der Aufzeichnung weiterer Messdaten wird empfohlen, zusätzliche Messungen zur Quantifizierung der Winkelabweichung durchzuführen und anschließend den Fehler in der Konfigurationsdatei der Sensoren zu korrigieren.

Bei der Berechnung der Metriken für die Verarbeitungsebene „Punktwolken“ werden für viele Metriken Occupancy Grids erstellt. Die Zellen der einzelnen Scan Grids nehmen dabei nur binäre Werte an, bevor aus den Scan Grids ein mit probabilistischen Zellwerten gefülltes Occupancy Grid erstellt wird. Eine Weiterentwicklung in diesem Bereich ist möglich, indem mittels eines Algorithmus direkt die probabilistischen Occupancy Grids erstellt werden, ohne zuvor Scan Grids zu erstellen. Hierbei muss die Verdeckung der einzelnen Zellen bezüglich beider Lidar-Sensoren berücksichtigt werden, falls sich ein Objekt zwischen einer Zelle und den Lidar-Sensoren befindet.

Eine zusätzliche Weiterentwicklungsmöglichkeit für zukünftige Bewertungen von Punktwolken stellt die OPDF- bzw. UPDF-Metrik dar. Der Suchradius für das Suchen nach belegten bzw. freien Zellen im jeweils anderen OG wird vor der Berechnung festgelegt. Um die Strahlaufweitung des Lidar-Sensors zu berücksichtigen, ist es möglich, einen dynamischen Suchradius einzuführen, der sich in größerer Entfernung vom Sensor gleichermaßen vergrößert.

Aus den zuvor genannten Gründen wird keine Bewertung der Verarbeitungsebenen „Segmentierung“ und „Klassifizierung“ durchgeführt und somit kann die Anwendbarkeit der hierfür gelisteten Metriken nicht überprüft werden. Eine Weiterentwicklung des implementierten Klassifizierungsalgorithmus ermöglicht die Überprüfung der Anwendbarkeit der Metriken zur Bewertung der Klassifizierung in einer nachfolgenden Arbeit.

In dieser Masterthesis wird festgestellt, dass der Trackingalgorithmus bei Verwendung von simulierten Eingangsdaten nicht alle Objekte in der Objektliste abspeichert. Es wird ein Workaround implementiert, durch den bei jedem Zeitschritt alle Objekte inklusive der gesamten Objekthistorie abgespeichert werden. Die dadurch generierte Objektliste wird bereits bei kurzen Szenarien groß und unübersichtlich. Durch eine Überarbeitung des Trackingalgorithmus in einer nachfolgenden Arbeit kann dieses Problem behoben werden.

Kurz vor Abschluss dieser Masterthesis wird die OSPA⁽²⁾-Metrik zur Bewertung von Trackingalgorithmen entwickelt.¹²⁴ Laut Beard et al. soll diese speziell bei der Berechnung von Szenarien mit vielen Objekten vorteilhaft sein. Ein Vergleich der OSPA⁽²⁾-Metrik mit den in dieser Masterthesis verwendeten OSPA-, OSPA-T- und OSPA-MT-Metriken kann zeigen, ob eine Anwendbarkeit der neuen Metrik für die Bewertung des Trackingalgorithmus möglich ist.

¹²⁴ Vgl. Beard, M. et al.: Performance Evaluation for Tracking Algorithms (2018).

Anhang

A.1 Metriken für die Verarbeitungsebene Punktwolken

Tabelle A-1: Metriken für das Szenario „i3 statisch Heck“ bei verschiedenen Abständen, Zellauflösung 0,04 m

Metrik	Abstand der Fahrzeuge zum Ego-Fahrzeug in Metern				
	10	20	30	40	50
OE	237,744	129,760	55,744	56,404	26,560
OE Ratio	0,471	0,387	0,453	0,548	0,632
1-OE Ratio	0,529	0,613	0,547	0,452	0,368
PCC	-0,026	0,114	0,031	-0,006	-0,012
BCC	-0,026	0,114	0,031	-0,006	-0,012
MS	4504,322	1610,414	4507,973	4437,598	967,924
MS Ratio	0,721	0,644	0,902	0,888	0,774
MS OCC	-1439,783	-719,098	-423,973	-496,127	-252,437
MS Ratio OCC	-6,728	-5,894	-8,153	-9,728	-10,097
OCR	0,000	0,053	0,063	0,000	0,000
FCR	0,981	0,966	0,993	0,991	0,982
OPDF ($r = 1$)	0,000	0,184	0,081	0,000	0,000
OPDF ($r = 2$)	0,000	0,411	0,213	0,000	0,098
OPDF ($r = 3$)	0,000	0,560	0,385	0,000	0,274
UPDF ($r = 1$)	0,986	0,976	0,995	0,994	0,988
UPDF ($r = 2$)	0,993	0,988	0,997	0,997	0,994
UPDF ($r = 3$)	0,995	0,992	0,998	0,998	0,996
Hausdorff-Distanz in m	3,292	0,680	2,572	1,871	0,988
Schwerpunkt-Distanz in m	0,949	0,218	0,240	0,991	0,420
PC2PC-Distanz in m	152,868	21,649	18,529	45,464	15,080

Tabelle A-2: Metriken für das Szenario „i3 statisch Front“ bei verschiedenen Abständen, Zellauflösung 0,04 m

Metrik	Abstand der Fahrzeuge zum Ego-Fahrzeug in Metern			
	20	30	40	50
OE	106,720	61,392	50,284	28,472
OE Ratio	0,430	0,379	0,513	0,678
1-OE Ratio	0,570	0,621	0,487	0,322
PCC	0,197	0,016	0,065	-0,007
BCC	0,197	0,016	0,065	-0,007
MS	1831,944	4460,933	5727,339	2205,181
MS Ratio	0,733	0,892	0,916	0,882
MS OCC	-521,958	-463,989	-464,019	-262,356
MS Ratio OCC	-5,019	-8,923	-10,087	-10,091
OCR	0,175	0,000	0,000	0,000
FCR	0,974	0,991	0,993	0,991
OPDF ($r = 1$)	0,278	0,098	0,000	0,000
OPDF ($r = 2$)	0,467	0,281	0,000	0,000
OPDF ($r = 3$)	0,576	0,450	0,195	0,117
UPDF ($r = 1$)	0,982	0,994	0,995	0,993
UPDF ($r = 2$)	0,991	0,997	0,998	0,997
UPDF ($r = 3$)	0,994	0,998	0,998	0,998
Hausdorff-Distanz in m	0,518	2,455	2,770	1,230
Schwerpunkt-Distanz in m	0,192	0,311	0,519	0,666
PC2PC-Distanz in m	21,076	24,096	26,521	21,219

Tabelle A-3: Metriken für das Szenario „i3 statisch Seite“ bei verschiedenen Abständen, Zellauflösung 0,04 m

Metrik	Abstand der Fahrzeuge zum Ego-Fahrzeug in Metern		
	10	30	40
OE	215,168	83,892	81,104
OE Ratio	0,439	0,344	0,365
1-OE Ratio	0,561	0,656	0,635
PCC	0,393	0,260	-0,006
BCC	0,393	0,260	-0,006
MS	5304,816	8933,038	8798,061
MS Ratio	0,849	0,953	0,938
MS OCC	-627,152	-321,885	-468,323
MS Ratio OCC	-2,570	-4,024	-7,318
OCR	0,374	0,256	0,000
FCR	0,980	0,996	0,996
OPDF ($r = 1$)	0,519	0,347	0,047
OPDF ($r = 2$)	0,721	0,523	0,197
OPDF ($r = 3$)	0,808	0,666	0,366
UPDF ($r = 1$)	0,986	0,997	0,997
UPDF ($r = 2$)	0,993	0,998	0,999
UPDF ($r = 3$)	0,995	0,999	0,999
Hausdorff-Distanz in m	1,452	1,535	1,547
Schwerpunkt-Distanz in m	0,111	0,198	0,538
PC2PC-Distanz in m	36,402	16,333	15,287

Tabelle A-4: Metriken für das Szenario „parkende Autos“, Zellauflösung 0,04 m

Metrik	Ergebnis der Metrik
OE	780,876
OE Ratio	0,349
1-OE Ratio	0,651
PCC	0,269
BCC	0,269
MS	685691,984
MS Ratio	0,992
MS OCC	-4328,103
MS Ratio OCC	-5,825
OCR	0,329
FCR	0,999
OPDF ($r = 1$)	0,433
OPDF ($r = 2$)	0,604
OPDF ($r = 3$)	0,695
UPDF ($r = 1$)	0,999
UPDF ($r = 2$)	Nicht berechnet
UPDF ($r = 3$)	Nicht berechnet
Hausdorff-Distanz in m	7,381
Schwerpunkt-Distanz in m	2,266
PC2PC-Distanz in m	205,958

Tabelle A-5: Zeitplan Masterthesis 699/18

Masterthesis 699/18 Sebastian Huch																																
Soll	X Ist	KW	19	20	21	22	23	24	25	26	27	28	29	30	31	August				September				Oktober				November				
VTD Schulung Vires Bad Aibling			X																													
Literaturrecherche (Lidar, Metriken)				X	X	X	X	X	X	X	X	X	X	X	X																	
Funktionsweise Lidar				X	X	X	X	X	X	X	X	X	X	X	X																	
Literatur zum Aufbau und zur Funktionsweise des Lidar-Sensor-Modells				X	X	X	X	X	X	X	X	X	X	X	X																	
Metriken für Punktwolken, Segmentierung, Tracking und Klassifizierung				X	X	X	X	X	X	X	X	X	X	X	X																	
Metriken																																
Auswahl geeigneter Metriken																																
Anpassung der ausgewählten Metriken nach Zwischenkolloquium																																
Zwischenkolloquium																																
Vorbereitung Präsentation Zwischenkolloquium																																
Zwischenkolloquium																																
Bewertung der vorhandenen Verarbeitungskette																																
Einarbeitung FMEA																																
FMEA durchführen																																
Gesamt-Metrik																																
Gesamt-Metrik entwickeln																																
Exemplarische Berechnung																																
Definition der zu testenden Szenarien																																
Durchführen von Versuchsfahrten in Griesheim																																
Ers tellen der Szenarien in VTD																																
Auswerten der Szenarien & Metriken anwenden																																
Interpretation der Ergebnisse																																
Ausarbeitung																																
Ausarbeitung schreiben																																
Ausarbeitung Korrektur																																
Ausarbeitung Abgabe																																
Abschlusskolloquium																																
Präsentation vorbereiten																																
Abschlusskolloquium																																

Literaturverzeichnis

Ackermann, S. M.: Masterthesis, Systematische Untersuchung von Radar Tracking-Algorithmen (2017)

Ackermann, Stefan M.: Systematische Untersuchung von Radar Tracking-Algorithmen, Masterthesis
Technische Universität Darmstadt, Darmstadt, 2017

Aeberhard, M.: Dissertation, Object-Level Fusion for Surround Environment Perception (2017)

Aeberhard, Michael: Object-Level Fusion for Surround Environment Perception in Automated Driving Applications, Dissertation
Technische Universität Dortmund, Dortmund, 2017

Aijazi, A. et al.: Segmentation Based Classification (2013)

Aijazi, Ahmad; Checchin, Paul; Trassoudaine, Laurent: Segmentation Based Classification of 3D Urban Point Clouds, in: Remote Sensing (4), Jahrgang 5, S. 1624–1650, 2013

Babahajiani, P. et al.: Urban 3D segmentation (2017)

Babahajiani, Pouria; Fan, Lixin; Kämäräinen, Joni-Kristian; Gabbouj, Moncef: Urban 3D segmentation and modelling from street view images and LiDAR point clouds, in: Machine Vision and Applications (7), Jahrgang 28, S. 679–694, 2017

Baizid, K. et al.: Vector Maps: A Lightweight and Accurate Map Format for Multi-robot Systems (2016)

Baizid, Khelifa; Lozenguez, Guillaume; Fabresse, Luc; Bouraqadi, Noury: Vector Maps: A Lightweight and Accurate Map Format for Multi-robot Systems, in: Kubota, Naoyuki et al. (Hrsg.): Intelligent Robotics and Applications, Imprint: Springer, Cham, 2016

Baizid, K.: Barons and Pearson Cross Correlation Coefficient (2018)

Baizid, Khelifa: Difference between Barons and Pearson Cross Correlation Coefficient, 2018

Balaguer, B. et al.: Evaluating maps from RoboCup (2009)

Balaguer, Benjamin; Balakirsky, Stephen; Carpin, Stefano; Visser, Arnoud: Evaluating maps produced by urban search and rescue robots, in: Autonomous Robots (4), Jahrgang 27, S. 449–464, 2009

Baron, R. J.: Mechanisms of human facial recognition (1981)

Baron, Robert J.: Mechanisms of human facial recognition, in: International Journal of Man-Machine Studies (2), Jahrgang 15, S. 137–178, 1981

Beard, M. et al.: Performance Evaluation for Tracking Algorithms (2018)

Beard, Michael; Vo, Ba-Tuong; Vo, Ba-Ngu: Performance Evaluation for Large-scale Multi-target Tracking Algorithms, in: 2018 21st International Conference on Information Fusion (FUSION), S. 1575–1581, 2018

Birk, A.: Learning Geometric Concepts with an Evolutionary Algorithm

Birk, Andreas: Learning Geometric Concepts with an Evolutionary Algorithm, in: In Proc. of The Fifth Annual Conference on Evolutionary Programming, Jahrgang 1996

Browning, B. et al.: Unmanned Ground Vehicle Lidar Simulation (2012)

Browning, Brett; Deschaud, Jean-Emmanuel; Prasser, David; Rander, Peter: 3D Mapping for high-fidelity unmanned ground vehicle lidar simulation, in: The International Journal of Robotics Research (12), Jahrgang 31, S. 1349–1376, 2012

Bundesministerium für Wirtschaft und Energie: Homepage Forschungsprojekt PEGASUS (2016)

Bundesministerium für Wirtschaft und Energie: Forschungsprojekt PEGASUS; <https://www.pegasusprojekt.de/de/home>, 2016, Zugriff 23.07.2018

Burgard, W.; Hebert, M.: World Modeling (2008)

Burgard, Wolfram; Hebert, Martial: World Modeling, in: Siciliano, Bruno; Khatib, Oussama (Hrsg.): Handbook of Robotics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008

Carlson, J. et al.: Conflict Metric as a Measure of Sensing Quality (2005)

Carlson, Jennifer; Murphy, Robin R.; Christopher, Svetlana; Casper, Jennifer: Conflict Metric as a Measure of Sensing Quality, in: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, S. 2032–2039, 2005

Carrilho, A. C. et al.: Quality assessment for automatic LiDAR data classification methods (2017)

Carrilho, André C.; Ivánová, Ivana; Galo, Mauricio: Quality assessment for automatic LiDAR data classification methods, in: Anais do XVIII Simpósio Brasileiro de Sensoriamento Remoto -SBSR, S. 6772–6779, 2017

Choe, Y. et al.: Fast Point Cloud Segmentation (2012)

Choe, Yungeun; Ahn, Seunguk; Chung, Myung J.: Fast Point Cloud Segmentation for an Intelligent Vehicle Using sweeping 2D laser scanners, in: 2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 2012

Collins, T. et al.: Occupancy Grid Mapping (2007)

Collins, Thomas; Collins, J. J.; Ryan, Conor: Occupancy Grid Mapping: An Empirical Evaluation, in: 2007 Mediterranean Conference on Control and Automation, 2007

Dewan, A. et al.: Classification for 3D LiDAR Data (2017)

Dewan, Ayush; Oliveira, Gabriel L.; Burgard, Wolfram: Deep Semantic Classification for 3D LiDAR Data, 2017

DIN Deutsches Institut für Normung e. V.: DIN ISO 8855 (2013)

DIN Deutsches Institut für Normung e. V.: DIN ISO 8855: Straßenfahrzeuge – Fahrzeugdynamik und Fahrverhalten – Begriffe (ISO 8855:2011), 2013

DIN Deutsches Institut für Normung e. V.: DIN EN 60812 (2015)

DIN Deutsches Institut für Normung e. V.: DIN EN 60812:2015-08: Fehlzustandsart- und -auswirkungsanalyse (FMEA) (IEC 56/1579/CD:2014), 2015

Douillard, B. et al.: Segmentation of 3D LIDAR point clouds (2011)

Douillard, B.; Underwood, J.; Kuntz, N.; Vlaskine, V.; Quadros, A.; Morton, P.; Frenkel, A.: On the segmentation of 3D LIDAR point clouds, in: 2011 IEEE International Conference on Robotics and Automation, S. 2798–2805, 2011

Douillard, B. et al.: Segmentation and Classification of 3D Point Clouds (2014)

Douillard, B.; Underwood, J.; Vlaskine, V.; Quadros, A.; Singh, S.: A Pipeline for the Segmentation and Classification of 3D Point Clouds, in: Khatib, Oussama; Kumar, Vijay; Sukhatme, Gaurav (Hrsg.): Experimental Robotics, Springer Tracts in Advanced Robotics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014

dpa: Uber-Unfall (2018)

dpa: Frau stirbt nach Unfall mit selbstfahrendem Auto von Uber; <http://www.sueddeutsche.de/news/wirtschaft/auto-frau-stirbt-nach-unfall-mit-selbstfahrendem-auto-von-uber-dpa.urn-newsml-dpa-com-20090101-180319-99-551280>, 2018, Zugriff 23.07.2018

Dubois, C.: Solid State LiDAR (2018)

Dubois, Chantelle: What Is Solid State LiDAR and Is It Faster, Cheaper, Better?; <https://www.al-lababoutcircuits.com/news/solid-state-lidar-faster-cheaper-better/>, 2018, Zugriff 24.07.2018

ECSEL Joint Undertaking: Homepage ENABLE-S3 (2016)

ECSEL Joint Undertaking: Homepage ENABLE-S3; <https://www.enable-s3.eu/>, 2016, Zugriff 23.07.2018

Fawcett, T.: An introduction to ROC analysis (2006)

Fawcett, Tom: An introduction to ROC analysis, in: Pattern Recognition Letters (8), Jahrgang 27, S. 861–874, 2006

Felzenszwalb, P. F.; Huttenlocher, D. P.: Efficient Graph-Based Image Segmentation (2004)

Felzenszwalb, Pedro F.; Huttenlocher, Daniel P.: Efficient Graph-Based Image Segmentation, in: International Journal of Computer Vision, 2004

Fernández-Madrigal, J.-A.: E-Mail-Verkehr bezüglich Map Score (2018)

Fernández-Madrigal, Juan-Antonio: Anwendung des Map Scores bei simulierten OGs, 2018

Fernández-Madrigal, J.-A.; Blanco Claraco, J. L.: Simultaneous localization and mapping (2013)

Fernández-Madrigal, Juan-Antonio; Blanco Claraco, José L.: Simultaneous localization and mapping for mobile robots, Information Science Reference, Hershey PA, 2013

Fischler, M. A.; Bolles, R. C.: Random Sample Consensus (1981)

Fischler, Martin A.; Bolles, Robert C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, in: Communications of the ACM (6), Jahrgang 24, 1981

GeneSys Elektronik GmbH: ADMA 3.0 Manual (2018)

GeneSys Elektronik GmbH: ADMA 3.0 Manual, 2018

Golovinskiy, A.; Funkhouser, T.: Min-Cut Based Segmentation of Point Clouds (2009)

Golovinskiy, Aleksey; Funkhouser, Thomas: Min-Cut Based Segmentation of Point Clouds, in: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, 2009

Gotzig, H.; Geduld, G.: LIDAR-Sensorik (2015)

Gotzig, Heinrich; Geduld, Georg: LIDAR-Sensorik, in: Winner, Hermann et al. (Hrsg.): Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015

Granström, K. et al.: Extended Object Tracking (2016)

Granström, Karl; Baum, Marcus; Reuter, Stephan: Extended Object Tracking: Introduction, Overview and Applications, in: Journal of Advances in Information Fusion (2), Jahrgang 12, 2016

Grewe, R. et al.: Accuracy of an Automotive Occupancy Grid (2012)

Grewe, Ralph; Komar, Matthias; Hohm, Andree; Lueke, Stefan; Winner, Hermann: Evaluation Method and Results for the Accuracy of an Automotive Occupancy Grid, in: 2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES), S. 19–24, 2012

Grilli, E. et al.: Point Clouds Segmentation And Classification Algorithms (2017)

Grilli, E.; Menna, F.; Remondino, F.: A Review Of Point Clouds Segmentation And Classification Algorithms, in: ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W3, S. 339–344, 2017

Gschwandtner, M. et al.: BlenSor: Blender Sensor Simulation Toolbox (2011)

Gschwandtner, Michael; Kwitt, Roland; Uhl, Andreas; Pree, Wolfgang: BlenSor: Blender Sensor Simulation Toolbox, in: Bebis, George et al. (Hrsg.): Advances in Visual Computing, Springer, Berlin, Heidelberg, 2011

Guyon, I. et al.: Feature Extraction (2006)

Guyon, Isabelle; Gunn, Steve; Nikravesh, Masoud; Zadeh, Lotfi A.: Feature Extraction, Studies in fuzziness and soft computing, Jahrgang 191, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006

Hakuli, S.; Krug, M.: Virtuelle Integration (2015)

Hakuli, Stephan; Krug, Markus: Virtuelle Integration, in: Winner, Hermann et al. (Hrsg.): Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015

Hanke, T. et al.: Generic Architecture for Simulation of ADAS Sensors (2015)

Hanke, Timo; Hirsenkorn, Nils; Dehlink, Bernhard; Rauch, Andreas; Raschofer, Ralph; Biebl, Erwin: Generic Architecture for Simulation of ADAS Sensors, in: 2015 16th International Radar Symposium (IRS), S. 125–130, 2015

Hanke, T. et al.: Validation of Virtual Point Cloud Data (2017)

Hanke, Timo; Schaermann, Alexander; Geiger, Matthias; Weiler, Konstantin; Hirsenkorn, Nils; Rauch, Andreas; Schneider, Stefan A.; Biebl, Erwin: Generation and Validation of Virtual Point

Cloud Data for Automated Driving Systems, in: 20th International Conference on Intelligent Transportation Systems, 2017

Hirsenkorn, N. et al.: A Non-Parametric Approach for Modeling Sensor Behavior (2015)

Hirsenkorn, N.; Hanke, T.; Rauch, A.; Dehlink, B.; Rasshofer, R.; Biebl, E.: A Non-Parametric Approach for Modeling Sensor Behavior, in: 2015 16th International Radar Symposium (IRS), S. 131–136, 2015

Hirsenkorn, N. et al.: Virtual sensor models for real-time applications (2016)

Hirsenkorn, Nils; Hanke, Timo; Rauch, Andreas; Dehlink, Bernhard; Rasshofer, Ralph; Biebl, Erwin: Virtual sensor models for real-time applications, in: Advances in Radio Science, Jahrgang 14, S. 31–37, 2016

Hoffman, J. R.; Mahler, R.: Multitarget Miss Distance via Optimal Assignment (2004)

Hoffman, J. R.; Mahler, R.P.S.: Multitarget Miss Distance via Optimal Assignment, in: IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans (3), Jahrgang 34, S. 327–336, 2004

Holder, M. et al.: Requirements for a Lidar Sensor Model (2018)

Holder, Martin; Rosenberger, Philipp; Bert, Felix; Winner, Hermann: Data-driven Derivation of Requirements for a Lidar Sensor Model, in: Graz Symposium Virtual Vehicle 2018, 2018

Ibeo Automotive Systems GmbH: Operating Manual ibeo LUX 2010® (2014)

Ibeo Automotive Systems GmbH: Operating Manual ibeo LUX 2010® Laserscanner, 2014

Ilic, V. et al.: ADAS Modules in Virtual Environments (2018)

Ilic, Velibor; Marijan, Malisa; Mehmed, Ayhan; Antlanger, Moritz: Development of Sensor Fusion Based ADAS Modules in Virtual Environments, in: 2018 Zooming Innovation in Consumer Technologies Conference (ZINC), S. 88–91, 2018

Jaccard, P.: Distribution of the Flora (1912)

Jaccard, Paul: The Distribution of the Flora in the Alpine Zone, in: The New Phytologist (2)XI, S. 37–50, 1912

Kernhof, J. et al.: LiDAR-Sensorsystem (2018)

Kernhof, Jürgen; Leuckfeld, Jan; Tavano, Guisepppe: LiDAR-Sensorsystem für automatisiertes und autonomes Fahren, in: Tille, Thomas (Hrsg.): Automobil-Sensorik 2, Springer Vieweg, Berlin, Heidelberg, 2018

Kim, B. et al.: Robust object segmentation using a multi-layer laser scanner (2014)

Kim, Beomseong; Choi, Baehoon; Yoo, Minkyun; Kim, Hyunju; Kim, Euntai: Robust object segmentation using a multi-layer laser scanner, in: Sensors (Basel, Switzerland) (11), Jahrgang 14, S. 20400–20418, 2014

Kim, B. et al.: Pedestrian/Vehicle Detection Using a 2.5-D Multi-Layer Laser Scanner (2016)

Kim, Beomseong; Choi, Baehoon; Park, Seongkeun; Kim, Hyunju; Kim, Euntai: Pedestrian/Vehicle Detection Using a 2.5-D Multi-Layer Laser Scanner, in: IEEE Sensors Journal (2), Jahrgang 16, S. 400–408, 2016

Klasing, K. et al.: A clustering method for efficient segmentation (2008)

Klasing, Klaas; Wollherr, Dirk; Buss, Martin: A clustering method for efficient segmentation of 3D laser data, in: 2008 IEEE International Conference on Robotics and Automation, S. 4043–4048, 2008

Knerr, J.: Masterthesis, Entwicklung eines LiDAR-Tracking-Algorithmus für Sensormodelle (2018)

Knerr, Jonathan: Entwicklung eines LiDAR-Tracking-Algorithmus für Sensormodelle, Masterthesis Technische Universität Darmstadt, Darmstadt, 2018

Lee, D.: Map-building and exploration strategies (1996)

Lee, David: The map-building and exploration strategies of a simple sonar-equipped robot, Distinguished dissertations in computer science, Cambridge University Press, Cambridge, 1996

Martin, M. C.; Moravec, H. P.: Robot Evidence Grids (1996)

Martin, Martin C.; Moravec, Hans P.: Robot Evidence Grids, 1996

Morton, P. et al.: Evaluation of dynamic object tracking (2011)

Morton, Peter; Douillard, Bertrand; Underwood, James: An evaluation of dynamic object tracking with 3D LIDAR, in: Proceedings of Australasian Conference on Robotics and Automation, 2011

Ngo, A.: Masterthesis, Simulationsbasiertes Testen und Entwickeln von Objekterkennungsalgorithmen (2018)

Ngo, Anthony: Erweiterung eines LiDAR Sensormodells für simulationsbasiertes Testen und Entwickeln von Objekterkennungsalgorithmen, Masterthesis
Technische Universität Darmstadt, Darmstadt, 2018

Nguyen, A.; Le, B.: 3D Point Cloud Segmentation (2013)

Nguyen, Anh; Le, Bac: 3D Point Cloud Segmentation: A survey, in: 2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM), S. 225–230, 2013

O’Sullivan, S.: Master’s Thesis, Empirical Evaluation Of Map Building Methodologies (2003)

O’Sullivan, Shane: An Empirical Evaluation Of Map Building Methodologies in Mobile Robotics Using The Feature Prediction Sonar Noise Filter And Metric Grid Map Benchmarking Suite, Master’s Thesis
University of Limerick, 2003

Oberkampff, W. L.; Barone, M. F.: Measures of agreement between computation and experiment (2006)

Oberkampff, William L.; Barone, Matthew F.: Measures of agreement between computation and experiment, in: Journal of Computational Physics (1), Jahrgang 217, S. 5–36, 2006

Oberkampff, W. L.; Trucano, T. G.: Verification and validation benchmarks (2008)

Oberkampff, William L.; Trucano, Timothy G.: Verification and validation benchmarks, in: Nuclear Engineering and Design (3), Jahrgang 238, S. 716–743, 2008

Papadimitriou, C.; Steiglitz, K.: Combinatorial optimization (1998)

Papadimitriou, Christos; Steiglitz, Kenneth: Combinatorial optimization, Dover Publ, Mineola, N.Y., 1998

Pieringer, C.: Masterarbeit, Modellierung des Fahrzeugumfelds mit Occupancy Grids (2013)

Pieringer, Christian: Modellierung des Fahrzeugumfelds mit Occupancy Grids, Masterarbeit
Universität Passau, Passau, 2013

Powers, D. M.: Receiver Operating Characteristics (2007)

Powers, David M. W.: Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation, 2007

Provost, F.; Domingos, P.: Improving Probability Estimation Trees (2000)

Provost, Foster; Domingos, Pedro: Well-Trained PETs: Improving Probability Estimation Trees, 2000

Rahmathullah, A. S. et al.: Multidimensional assignment distance (2016)

Rahmathullah, Abu S.; García-Fernández, Ángel F.; Svensson, Lennart: A metric on the space of finite sets of trajectories for evaluation of multi-target tracking algorithms, 2016

Rahmathullah, A. S. et al.: Generalized optimal sub-pattern assignment metric (2017)

Rahmathullah, Abu S.; García-Fernández, Ángel F.; Svensson, Lennart: Generalized optimal sub-pattern assignment metric, in: Proceedings of the 20th International Conference on Information Fusion (Fusion), 2017, S. 1–8, 2017

Ristic, B. et al.: A Metric for Performance Evaluation of Multi-Target Tracking Algorithms (2011)

Ristic, B.; Ba-Ngu Vo; Clark, D.; Ba-Tuong Vo: A Metric for Performance Evaluation of Multi-Target Tracking Algorithms, in: IEEE Transactions on Signal Processing (7), Jahrgang 59, S. 3452–3457, 2011

Rosenberger, P. et al.: Analysis of Real World Sensor Behavior (2018)

Rosenberger, Philipp; Holder, Martin; Zirulnik, Marina; Winner, Hermann: Analysis of Real World Sensor Behavior for Rising Fidelity of Physically Based Lidar Sensor Models, in: IEEE Intelligent Vehicles Symposium (IV), S. 611–616, 2018

Ross, P. E.: Solid-State Lidar (2018)

Ross, Philip E.: BMW Will Use Solid-State Lidar From Innoviz; <https://spectrum.ieee.org/cars-that-think/transportation/sensors/embargoed-until-7am-edt-thursday-25-april-bmw-will-use-solidstate-lidar-from-innoviz>, 2018, Zugriff 24.07.2018

Roth, E. et al.: Analysis and Validation of Perception Sensor Models (2011)

Roth, Erwin; Dirndorfer, Tobias J.; Knoll, Alois; Neumann-Cosel, Kilian von; Ganslmeier, Thomas; Kern, Andreas; Fischer, Marc-Oliver: Analysis and Validation of Perception Sensor Models in an Integrated Vehicle and Environment Simulation, in: 2011 22nd Enhanced Safety of Vehicle Conference, 2011

SAE International: J3016 (2018)

SAE International: 1: J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, 2018

Sandner, T. et al.: MEMS Mirror Array (2008)

Sandner, Thilo; Wildenhain, Michael; Klose, Thomas; Schenk, Harald; Schwarzer, Stefan; Hinkov, Vladimir; Hofler, Heinrich; Wolfelschneider, Harald: 3D Imaging Using Resonant Large-Aperture MEMS Mirror Arrays and Laser Distance Measurement, in: 2008 IEEELEOS International Conference on Optical MEMS and Nanophotonics, S. 78–79, 2008

Schaermann, A. et al.: Validation of Vehicle Environment Sensor Models (2017)

Schaermann, Alexander; Rauch, Andreas; Hirsenkorn, Nils; Hanke, Timo; Rasshofer, Ralph; Biebl, Erwin: Validation of Vehicle Environment Sensor Models, in: 2017 IEEE Intelligent Vehicles Symposium, 2017

Schuhmacher, D. et al.: Metric for Performance Evaluation of Multi-Object Filters (2008)

Schuhmacher, Dominic; Vo, Ba-Tuong; Vo, Ba-Ngu: A Consistent Metric for Performance Evaluation of Multi-Object Filters, in: IEEE Transactions on Signal Processing (8), Jahrgang 56, S. 3447–3457, 2008

Shin, M.-O. et al.: Segmentation of 3-D Point Clouds (2017)

Shin, Myung-Ok; Oh, Gyu-Min; Kim, Seong-Woo; Seo, Seung-Woo: Real-Time and Accurate Segmentation of 3-D Point Clouds Based on Gaussian Process Regression, in: IEEE Transactions on Intelligent Transportation Systems (12), Jahrgang 18, S. 3363–3377, 2017

Stäcker, L. S.: Masterthesis, Informationssystem zum Segmentieren und Tracking von Objekten (2017)

Stäcker, Lukas S.: Entwicklung eines Informationssystems zum Segmentieren und Tracking von Objekten auf Basis von LiDAR-Sensordaten für Autonomes Fahren, Masterthesis Technische Universität Darmstadt, Darmstadt, 2017

Stäuber, P.: Radiusberechnung nach WGS84 (2018)

Stäuber, Philipp: Radiusberechnung nach WGS84, 2018

Velodyne LiDAR Inc.: HDL-64E S3 (2018)

Velodyne LiDAR Inc.: HDL-64E S3; http://velodynelidar.com/docs/datasheet/63-9194_Rev-J_HDL-64E_S3_Spec%20Sheet%20Web.pdf, 2018, Zugriff 24.07.2018

VIRES Simulationstechnologie GmbH: Virtual Test Drive (2018)

VIRES Simulationstechnologie GmbH: Virtual Test Drive; <https://vires.com/vtd-vires-virtual-test-drive/>, 2018, Zugriff 07.09.2018

Vu, T. A.; Evans, R.: Optimal Subpattern Assignment Metric for Multiple Tracks (2018)

Vu, Tuyet A. T.; Evans, Rob: Optimal Subpattern Assignment Metric for Multiple Tracks (OSPAMT Metric), 2018

Wachenfeld, W.; Winner, H.: Die Freigabe des autonomen Fahrens (2015)

Wachenfeld, Walther; Winner, Hermann: Die Freigabe des autonomen Fahrens, in: Maurer, Markus (Hrsg.): Autonomes Fahren, Springer, Berlin, Heidelberg, 2015

Wandinger, U.: Introduction to Lidar (2005)

Wandinger, Ulla: Introduction to Lidar, in: Weitkamp, Claus (Hrsg.): Lidar, Springer Science+Business Media Inc., New York, NY, 2005

Winner, H. et al.: Handbuch Fahrerassistenzsysteme (2015)

Winner, Hermann; Hakuli, Stephan; Lotz, Felix; Singer, Christina (Hrsg.) Handbuch Fahrerassistenzsysteme, 3. Auflage, Springer Vieweg, Wiesbaden, 2015

Winner, H.: Vorlesungsunterlagen Mechatronik und Assistenzsysteme im Automobil (2017)

Winner, Hermann: Vorlesungsunterlagen Mechatronik und Assistenzsysteme im Automobil - Lidar-sensorik, 2017

Yokoyama, H. et al.: Detection and Classification of Pole-like Objects (2013)

Yokoyama, Hiroki; Date, Hiroaki; Kanai, Satoshi; Takeda, Hiroshi: Detection and Classification of Pole-like Objects from Mobile Laser Scanning Data of Urban Environments, in: International Journal of CAD/CAM (1), Jahrgang 13, S. 1–10, 2013

Zhou, Y.; Tuzel, O.: VoxelNet (2017)

Zhou, Yin; Tuzel, Oncel: VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection, 2017